Tutorials: Imperfect Surfaces, 5 Minute Lightning, Modeling Complex Objects

DEBUT ISSUE

KEYFRAME MAGAZINE

THE PUBLICATION FORMERLY KNOWN AS LIGHTWAVIN'

Marvelous Anime: Silver Surfer

Joe's Digital Bar & Grill brings the comic to life

Visit with "Rocketman" Mark Thompson of The Big Machine

Place with computer magazines

Issue #6 May/June 1998
$9.95 US $14.95 Canada

Reviews: Inside LightWave 3D, Dark Tree Textures, Film Grain, Motion Pack, Relativity

Particle Storm 2.0

Available in June! This long awaited upgrade to the world's most popular LightWave 3D plug-in will be debuted at the NewMedia '98 Expo in Toronto, Ontario, Canada May 13-15. NAB attendees can preview Particle Storm 2.0 at NewTek's booth. Available for Intel, Alpha, Mac, Sun, and SGI, Particle Storm 2.0 will include many new features such as:

Complex Particles: No more need to use only single-point polygons! Particles can be any complex shape you want. Size changes and rotations are now possible. Why settle for just dust in the wind when you can have leaves or even cows feel the breeze as well! Strand particles are a special breed, making effects like silly string and flowing hair a cut above the rest. Turn the polygons of your object into particles and watch your object explode.

Greater Integration with Layout: A simulation can now be controlled from within Layout as well as from the main Particle Storm interface! This makes it simple to tweak parameters and see the results right in Layout.

Animatable Parameters: The first version gave you parenting, and the new version goes one better! Now you can animate almost every parameter available. Decay the birthrate, make collisions bouncier and bouncier, turn a calm breeze into a strong gust or even a tornado. And best of all, it's all done automatically!

Enveloping: Now the animator can envelope most of Particle Storm's parameters to create variable particle effects.

Redesigned GUI: The radical new GUI makes it easier than ever to create the effects you want. Supporting OpenGL (QuickDraw3D for MacOS) and Lightwave Panel's architecture maintains the look and feel of Particle Storm across all platforms. You can instantly jump to the data you want to without having to go through layers of interfaces. Go from one part of the GUI to another with ease.

Particle Spawning: Now, particles can emit other particles due to collisions, speed limits, or any arbitrary time you want.

Collision Avoidance: Birds and bees can now steer clear of obstacles, and your particles can avoid each other. Use with flocking behavior to create even more realistic flocking behavior than ever before!

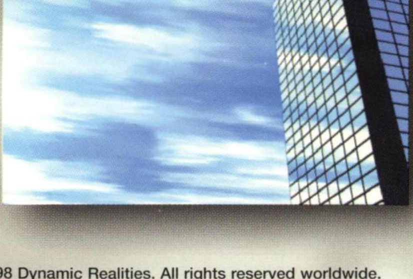
Fractalizer

Fractalizer is a new surfacing utility for LightWave that surpasses LightWave's current fractal shader, Fractal Noise. This plug-in is great for making realistic surfaces, even ones previously difficult or impossible to achieve! Create dirty objects, rust, clouds, water, plasma, electric sparks, fire, and many more natural surfaces for your LightWave objects. Available now for Intel, Alpha, Mac, Sun, and SGI, the features that make Fractalizer so special are:

Five Color Spectrum: Allows easy and intuitive changing of the cloud size and the contrast between multiple colors. Almost an infinite number of colors and combinations can be achieved!

Radial Falloff and Fractal Opacity: These effects can create many unique looks, like burned out holes.

Fractal Morphing: Allows obtaining the effect of a changing color or texture without tedious object by object morphing.



Dynamic Realities

1900A Pewaukee Road Waukesha, WI 53188 USA
Ph. 414-549-2200 Fax 414-549-2225 Sales 800-944-5520
For more info visit www.dynamic-realities.com

All products mentioned are trademarks of their respective companies. © 1998 Dynamic Realities. All rights reserved worldwide.

New Plug-ins for LightWave 3D

Come visit us!

at NAB in NewTek's booth,
NewMedia Expo, Toronto, booth 718,
and at LENY '98.

LumeTools

Available in June! From the studios of SoftImage animators came *Riven*, the sequel to the award winning game, *Myst*. Those same animators used LumeTools for the stunning surfaces, water, skies, and lighting. Now, LumeTools is available for LightWave animators. LumeTools can be purchased as an entire collection, or in related sets, listed below.

LumeLandscape Includes *Landscape* for mapping textures to terrain models using slope, height, and noise functions, *Mist* for creating truly layered fog with depth fading, and *Façade*, which places 2D cardboard cutouts in the scene and rotates them with the camera. A great memory and time saver!

LumeWater Comes with *Ocean* to simulate animateable and looping waves on large water bodies, *Water* for modeling the physics of the reflective, transparency, and coloration properties of water, as well as letting objects know when and where they are wet and applying textures accordingly, *Submerge* for underwater depth fading effects, *Wet* to facilitate the creation of wet objects with saturated colors and specular highlights, and *Stain* for confining textures with the boundaries of other objects.

LumeLight Includes *Glare* to provide realistic glares and flares from any bright source, not just lights, and models the way the human eye scatters light. *Translucency* creates truly translucent textures with blurred transparency and back-lighting, scattering effects. *Glow* provides internal lighting to an object, *Night* simulates the eye's rods and cones, making colors appear desaturated in dim lighting (i.e. moonlight), *Beam* gives a basic atmospheric effect around lights without shadow calculations for high-speed renderings, and *Illumination* provides a physically accurate lighting model for light sources.

LumeMatter Includes *Metal* and *Glass* for modeling the reflective and transparent properties of these surfaces, including lighting shifts near edges, reflection blurring, coloration, and more, and *Edge* for special edging effects like a fuzzy mouse, a notched peg, a blurry ghost, or a jagged rock.

LumeWorkbench Provides simple yet important tools, like *WrapAround* for a 360 degree rendering, *Distortion* for simulating wide-angle lenses, *BumpCapture* to create bump-maps right in LightWave, and *Adjustments* for adjusting the coloration, brightness, contrast, hue, etc. of images. Can be applied on a per-object basis.

Created using LumeWater

Dynamic Realities' software is always in stock at

SAFE HARBOR
COMPUTERS

1-800-544-6599
www.sharbor.com

Reader Service #101

Subscribe Now!

- Unbiased Reviews
- Useful Tutorials
- Experienced Writers
- Industry Secrets
- Focused Exclusively on LightWave and Inspire

LIGHTWAVIN'
The Magazine For LightWave Animators
SNAKING OBJECTS ALONG A PATH
REVIEWS
Organic Modeling & Animation
LW 3D Plug-in Power
Leny CD-ROM Vol 2
LightWave Lab 2
Steamer Video
Genesis VFX
Pyromania II
Realitools
Key Pro
LIQUID METAL
Create T2 Type Effects!
LIGHTW
The Magazine For Light
REVIEWS
Reel Explosions CD
Tutorvision 3DCD
Surface Effectors
Dinosaurs ROM
Reel Fire CD
Super Glow
Leny CD
&
More
NEW COLUMN
Ranting!
INSIDE STEAMER
Smoke, Fire & Explosions!

- Back Issues Available. Call 800-778-WAVE or 805-264-4991 for pricing and availability.

Editor
Patrik Beck pat@keyframemag.com

Technical Editors
David Warner
Anupam Das

Advertising Contact
Catherine Baus

Contributing Writers
Anupam Das
Tiffani Banaszak
Doug Kelly
Dave Warner
Jay Watson
William Capozzi
Brenden Mecleary

Offices
2763 West Avenue L, Suite 162
Lancaster, California 93536 USA
Phone 805-264-4991
Fax 805-264-4998
Email: keyframe@keyframemag.com

Subscription Information
888-778-WAVE
805-264-4991
805-264-4998 Fax
Electronic submission form available at
http://www.keyframemag.com

Keyframe Magazine is looking for writers! If you have a good working knowledge of LightWave 3D or Inspire, and would like to contribute tutorials, news, or product reviews to this publication, please contact Patrik Beck.

Letters to the Editor can be mailed to the address above, attention Patrik Beck, or emailed to pat@keyframemag.com. Press releases and news items can be mailed to the attention of Ear to the Wall, or emailed to news@keyframemag.com.

KeyFrame Magazine is published 6 times annually by DMG Publishing. ©1998 by DMG Publishing. All rights reserved. Reproduction in part or in whole, without written permission, is strictly prohibited. Printed in the USA

Cover Image courtesy of Joe's Digital Bar and Grill. Image TM and © 1997 Saban. Silver Surfer is the exclusive property of the Marvel Entertainment Group Inc. All rights reserved.

Image created using LightWave 3D, Cel Shader, and SuperCel Shader for Galactus, Particle Storm and SuperGlow for the eye effects, and traditional hand-drawn cel animation for Silver Surfer.

KEYFRAME MAGAZINE

Issue #6 May/June 1998

In Every Issue

Editorial Change Is Good
by Patrik Beck Page 4
Feature Article: The Revolution Continues
by Patrik Beck Page 6
Ear to the Wall News, Press Releases, and Events Page 8
On Topic Your LW Q&As
With Dave Warner Page 11
Wave Surfing LightWave-focused websites Page 14
Profiler A Visit with Mark Thompson of The Big Machine Page 26
LightWave MultiPlatform Macintosh details
by Jason Goodman Page 28
Programmer's Hideaway Write your own LightWave plug-ins
by Brenden Mecleary Page 30
Take One Silver Surfer and Joe's Digital Bar & Grill Page 36
Final Cut Out with the old, in with the new
by Patrik Beck Page 48

Reviews You Can Use

Relativity Page 38
Inside LightWave 3D Page 40
NewTek Model Library Page 41
Dark Tree Textures Page 42
Film Grain Plug-In Page 43
Motion Pack Plug-In Page 44

Learning Curve: Tutorial Section

Perfectly Imperfect Surfaces by William Capozzi Page 16
5 Minute Miracle: Making a Lightning Bolt Page 18
Ducati: Modeling a Complex Object by James Hans Page 20
Character Shop by Doug Kelly Page 22
LightWave Mailing List Starter Kit by Dave Warner Page 24

• FACING A WORLD OF CHANGE, ONE KEYFRAME AT A TIME.

EDITORIAL

by Patrik Beck

Please allow me to introduce myself... My name is Patrik Beck, and I am the new Editor of this publication. Manny Coats, former owner and editor of *Lightwavin' Magazine*, has moved on, and I am stepping in to fill his shoes. You may already be familiar with my work. I have contributed to several magazines and books in the past, such as *LightWave Pro, Video Toaster User, NT Studio, NewTekniques,* and *Inside LightWave 3D*. If you come from an Amiga background, you may know that I wrote for nearly every Amiga specific publication there was (and there were surprisingly many). Becoming the Editor for a magazine is a logical next step for me.

Though I will be devoting much of my time to creating a great magazine, my full-time occupation is an animator. I use LightWave every day and every dollar I have made in the last several years was because of it. I could be working in Hollywood; I have had offers, but LightWave allows me to support my family in an area where I want to live doing something I love.

Change

The only thing that you can count on staying the same is that things keep changing. This is more true in the world of computers and 3D animation than anywhere else. Not only does it change, but it changes quickly. There is no such thing as a shelf-life when it comes to computer hardware or software.

LightWave is in a constant state of evolution. Since its first release, not only has it been growing in power and ability; it has been changing the world as well. There is a whole industry that owes much to the popularity of LightWave. Computer companies have grown strong by designing 'LightWave Workstations'. Hardware has been created specifically for compiling and playing back LightWave rendered images. There are even individuals that generated income from building and selling 3D models or programming simple utilities that made LightWave even more powerful.

In the last few years, we have seen LightWave grow from an industry secret, sitting on a Video Toaster System hidden in the back room of many production houses, to being an integral part of some of the biggest movies ever made. LightWave is now running on systems that are literally 1000 times more powerful then its original platform, and the rate of change keeps getting faster and faster. I do not know whether to be excited or afraid. Probably both.

I have been a LightWave user since before the Toaster was released (beta tester with board SN#8). Even though I use LightWave on a daily basis, and it is the heart of my profession; even though I have written about it in books and magazines; every time I open up the program I still feel like I am just scratching the surface of what it can do.

Lightwavin', more then any other publication, has always been a source of inside information and professional techniques. It gave voice to many people from whom you don't often hear. These may be people in the field who have figured out a technique to do something they needed for a specific game, or maybe programmers who explain why a plug-in does what it does and what is the best way to use it. They may not have had the polish of a professional writer, but their articles came from the heart of someone who believed that LightWave has the power to create anything a mind can imagine. There are some things that should change, but there are things that should stay the same, *but improved*.

Lightwavin', now renamed to *Keyframe Magazine,* will remain a unique voice. It will also grow and explore. I want this magazine to continue to be the place where the serious animator looks for fresh ideas. The place were we push the envelope. The place where we find the limits of what LightWave can do and how to surpass them. Whether you are an old-timer from the Amiga days or a novice with a fresh perspective, *Keyframe's* goal is to inform and inspire.

The founder of *Lightwavin'*, Manual Coats, is moving on, having contributed greatly to the LightWave community. Picking up where he has left off is a job I approach with both fear and excitement. The opportunity to become Editor of this magazine affords me the opportunity to move closer to the front of the LightWave roller coaster. I look forward to meeting new people and sharing ideas. More then anything, I look forward to the next chapter of the continuing story that is the ever changing world of LightWave.

Delay of Game

The last issue of *Lightwavin'/Keyframe* was delayed considerably because of the transfer of command. Of course, with every new enterprise, everything takes twice as long as you think or is promised. The issue in your hand was hurried out the door for two reasons. Firstly, we wanted to have the new name and format for NAB, and secondly, to establish a precedent of getting issues our in a timely manner. We will continue to do our best to get issues out on time.

Patrik Beck, Editor

TO ORDER CALL 800-544-6599
MONDAY-FRIDAY 9-6, CST
INFORMATION 414-548-8120 • FAX 414-548-8130
TECH SUPPORT/RMAs 414-548-8159 • 12-4 PM, Mon-Fri.

WE NEVER CLOSE • ORDER SECURELY ON-LINE
www.sharbor.com

CALL OUR Internet site www.sharbor.com

Shop our secure online store for a bigger selection and more information about each product at www.sharbor.com

MasterCard VISA DISCOVER NOVUS

Safe Harbor Computers –

Your #1 Source for Digital Video, 3D Animation, and Multimedia Products.

LIGHTWAVE 3D

LightWave 3D 5.5 – Powerful 3D software that unleashes your imagination and gives you the freedom to put creativity in motion!

LightWave 5.5, Mac	1675.00
LightWave 5.5 Intel/DEC	1799.00
LightWave 5-5.5 Upgrade, Intel	459.00
LightWave 4-5.5 Upgrade, Intel	629.00
LightWave 4-5.0 Upgrade, Amiga	289.00
LightWave 5-5.5 Upgrade, Mac	449.00

Call for Pricing on Toaster and Flyer Systems.

LIGHTWAVE PLUG-INS

Film Grain	139.00
Motion Pack, Intel/DEC	229.00

NEW! JUST IN!

Inspire 3D	Call

Call for School or Government discounts, Competitive or Cross Platform Upgrades!

NewTek

Check your video equipment anytime...anywhere with Calibar! Just tuck Calibar in your pocket alongside your ball point pen; it's small, lightweight, totally portable, and available at the touch of a button – the niftiest invention since the toaster! Calibar performs 24 test pattern functions, accurately.

NewTek

Calibar Test Signal Generator	**299.00***

**After instant $30 Safe Harbor rebate!*

TUTORIALS

VIDEO TAPES

Advanced Features, LightWave	39.99
Cinematic Visual Effects	39.99
Digital Cinematography	39.99
Flyer Advanced Techniques	39.99
LightWave 3D 5.0 Tutorial Bundle	139.99
LightWave 3D Modeler 1 & 2 Bundle	69.00
LightWave Lab Vol. 1 or 2	Call
LightWave Quickstart 5.5	39.99
Mastering Modeler	39.99
Organic Modeling & Animation	39.00
PhotoShop Essentials *New!*	39.99
Plug-In Power *New!*	39.00
Pro Flying Logos	39.99
Pyrotechniques	39.99
Steamer - Effects Tutorial	39.00
Surfacing Techniques	39.99
Spacecraft Modeling Design	39.99
Spacecraft Surfacing	39.99
Video Toaster Essentials	39.99

BOOKS

Flyer Mastery Guide Vol 2	125.00
Inside LightWave 3D 5.5	41.99
LightWave 5 Character Animation	44.00
LightWave Power Guide	44.00
PhotoReal F/X	42.00
The LightWave 3D Book	38.99

CD ROM TUTORIALS

TutorVision 3D CD	49.99
LENY CD	69.00
LENY CD Vol 2 Char Anim	45.00

NEW!

New from Tom Marlin Studios. Look no further! The name on this CD says it all! You get 252 textures, all seamless and tileable in JPeg format, PLUS 192 matching bump maps – all you can really use! Works with LightWave and other 2D/3D software packages.

Seamless Textures You Can Really Use	**89.00**

Leadtek

WinFast 3D L2300 PCI or AGP	169.00
WinFast 3D L2500 Gold	689.00

NEW LOWER PRICING!

Oxygen™ dp dynamic pictures

Oxygen 102, Intel	375.00
Oxygen 202, Intel	859.00
Oxygen 402, Intel	1699.00

Call us for Oxygen/LightWave Bundles!

LIGHTWAVE PLUG-IN/ LIGHTWAVE RELATED SOFTWARE

AVI Load	49.00
Boris FX 3.1	425.00
Con-Motion	Call
Fractalizer, Intel/Mac	129.00
Hollywood FX 3,0	359.00
Humanoid for LightWave	160.00
Light ROM Vol. 5 CD	39.00
Light ROM Gold CD	24.00
Lock & Key, Intel/Mac	129.00
MacroForm 2.0, Intel	245.00
MacroForm 2.0, DEC	279.00
Movie Models Aviation	39.00
Organica	149.00
Particle Storm 2.0	Call
Partycles	Call
PhotoShop	559.00
Pyromania Bundle	195.00
RealiTools, Intel/Mac	129.00
Reel Fire	315.00
Relativity	Call
Scene Machine, Intel	85.00
Solar System Kit/LW	93.00
Shaderman, Intel/Mac	139.00
Universal 3D CD	119.00
Vertilectric, Intel/DEC	75.99
Vertitilities, Intel/DEC	99.00
World Construction Set 3.0, Intel/Mac	649.00

Puppet Master – Amazing plug-in for LightWave! Created by Fori Owurowa, this product aids in character animation. Features: Allows you to pick points on your object by assigning them as separate groups and move them individually, enables you to cut your object into sections, and much more.....Definitely a must have! Requires LightWave 5.0 and up.

Puppet Master, Intel/Mac	**299.00**
DEC	**349.00**
Amiga	**159.00**

Puppet Master

Dab, Intel/DEC Alpha	89.00
Extract Audio, Intel	59.99
Fiber Factory 2, Intel/DEC	129.00
Particle Type, Intel	79.99
Point At, Intel/Mac	59.99
ProPak Bundle, Intel/DEC	199.00
Sparks, Intel	154.99
Wobbler, Intel/MAC	59.00

METROgrafx Animation and Software

Safe Harbor is dedicated to providing quality digital video, animation and multimedia products, technical expertise, and courteous, personalized service to our customers to maintain our impeccable reputation and total customer satisfaction.

Terms: POs accepted from schools and government agencies • Personal checks require 7 days to clear. • Defective products replaced promptly. RMA number required (call 414-548-8159) for all merchandise returns. Returns not accepted after 15 days. Returned products must be in original packaging, postage prepaid. Opened software not returnable. Shipping charges not refundable. Returns subject to a 15% restocking fee.• Not responsible for typos. Prices subject to change.

SAFE HARBOR COMPUTERS

W226 N900 EASTMOUND DR., WAUKESHA, WI 53186

Reader Service #102

The Revolution Continues...

This is the time of year when new products are being released one on top of the other. Debuts are announced at NAB and CEBIT, and people line up to get a peek at products in development. Two products that are going to revolutionize the LightWave market this year are just coming out: Inspire 3D, which will be shipping by the time this prints, and Particle Storm 2.0, which will be shown in beta at NAB and released at the NewMedia '98 show in Toronto in May.

by Patrik Beck

Do you need to be Inspired?

Inspire, NewTek's latest 3D offering, has a lot going for it. In fact, at first glance it hardly looks at all different from the full version of LightWave. Rather then ghosting out more advanced features, NewTek simply redesigned a few panels so that the few features that are missing are hard to spot.

Two of the most important limitations that may be deciding factors for people considering Inspire 3D for their main 3D package are the lack of networkability and no Screamer Net functions. Also, though you can still numerically enter the resolution, you can only render square pixels. All the panels with plug-in capability only have two slots. In Modeler, the biggest limitation is that there are only four layers to work with and there are no Boolean operations. (While most beginners probably will not be hampered by missing functions like 'lens flare streak density' or 'distance dissolve envelope', the one missing feature I do question is 'Depth of Field'.)

The truth is, NewTek did not trim much. There is an amazing amount of LightWave left in. There are still such advanced functions as Bones and MetaNURBS. The features that are gone will probably only be missed by people that are already using the full version. Inspire 3D is flexible enough that a determined soul can find a work-around for nearly any situation. Inspire 3D has nearly all the power of LightWave 5.5, and many of the missing features can be replaced with 3rd party plug-ins. For a casual user, hobbyist, or person just entering the field, Inspire 3D is not just a great first step, but a stepping stone to greater things.

Storming for New Particles

It's here, and it is about time! The most popular LightWave plug-in of all time, Particle Storm, is about to be released in version 2.0, and it has some wonderful new features. Most of these enhancements are the results of user feedback, both from the Hollywood motion picture industry and from the average user. Particle Storm in version 2.0 is about to make sparks all over again.

Some Background

Particle Storm is a Layout plug-in that gives LightWave the power to create particle type animation. Typical examples are smoke and dust, though with Particle Storm the particles may be replaced by other objects for effects like flying shell casings or swarming bees.

Particle Storm is based upon two separate plug-ins, PSCreate and PSPlay. With PSCreate, the user generates a single object that contains all the particles, and makes a separate file that contains the motion path for each point that is applied as a displacement map. The PSPlay allows you to load the particle object and its displacement file into Layout.

Particles can have attributes such as color and transparency assigned to them within the PScreate screen, and be modified to change with velocity, age, and collision. Particle density, birthrate, velocity, and lifespan can all be set. The particle path can be directed and perturbed by a number of effectors and forces defined by the user, many of which can be parented to objects currently loaded in Layout. Even NewTek thought Particle Storm was so impressive that a functional, though slightly crippled, Particle Storm Lite is included with the LightWave 5.5 upgrade. With all of this going for it, what more can you ask for? Turns out you can ask for a lot more. Users have inundated Dynamic Realities with requests and suggestions. The consequence of that feedback is an update that will reward current users for their patience and send non-users running for their speed dial to order.

What's New With Particle Storm 2.0

Pretty much everything that was in version 1.0 has been made better, from the interface on down. I looked at a beta version, which is very similar to what will be shipping not long after this issue is on the stands. More detailed information can be found on the Dynamic Realities web site (www.dynamic-realities.com). We will take an overview of some of the cool new features one at a time.

New User Interface

The interface is now much more intuitive and easier to navigate due to the integration of a tree-type control. This is shown in the screen shot (Fig 1). The particle preview window is now OpenGL shaded to make viewing collision objects much easier. Particle Storm's interface will be consistent across all platforms as well. Particle Storm 2.0 will be available for Intel, DEC Alpha, Power Macintosh, Sun, and SGI.

Strands

This is not the most important new feature, but it is the one everyone will be talking about. Particles streaming from an object being used as a fountain emitter can have the particles connected by a two point polygon (the results are similar to silly string or hair). The points of these two point polygons react exactly as other particles; they can be affected by wind, gravity, collision detection, and be colored with the particle shader. When loaded into Layout the strands can be textured the same as any other 1 or 2 point polygon. Hair can be grown over time and a future version promises to provide strand constraints giving you the capability to do curly hair.

Polygon Replacemnt

You can replace an object's polygons with particles. This is a great way of making objects explode into dust particles!

Envelopes

This probably is the most important new feature. It is now possible to vary nearly every value in the PSCreate plug-in screen over time. Envelopes can be edited directly without having to reference a layout item. Particle Storm parameters can be referenced to an object in the layout screen providing tighter integration between PSCreate and layout. Birthrate of particles can be tied to position of an object, or wind strength can rise and fall depending on the keyframed values of a referenced null. No need to jump back into PSCreate to resimulate as the new values are shown immediately in layout. You use this ONCE and you will wonder how you ever got along with out it.

Collision Avoidance

With this turned on, particles will try to veer around a collision object. This is useful in conjunction with flocking and swarming to create a group of animals that are following a leader and also veer away from obstacles.

Inter-Particle Avoidance

Because particles have no actual dimension, any attempt to fill a container with them will be futile. This makes volume effects like cascading waterfalls difficult. The Inter-Particle controller places a force field around all particles of the same particle group so that the particles have the effect of actually taking up space. This is the first time that the particles are interacting with each other instead of outside forces, and is sure to lead to some interesting applications, such as gaseous and liquid effects.

New Shading Options

A number of changes have been made to the Particle Shader. Individual selection of red, green, blue, and alpha means users may now choose which channels of a particle color to use for their particle shading.

Particle Sub-objects

Instead of using the Particle Storm post process to replace the particles with objects, you can create the Particle Storm object containing specified sub objects. The sub object can then take on many of the attributes available to particles, including shading and transparency.

Particle Spawning

Particles can generate more particles based on collisions, particle age, and speed.

Sticky Particles

Particles that collide with collision objects can stick to the object's geometry.

Particle Assembly

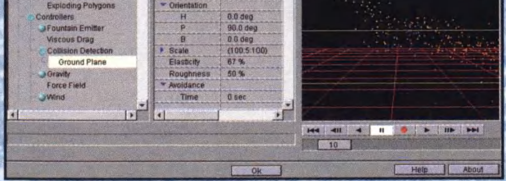
This is great for bringing particles together to form shapes or particle configurations such as a flock of birds that fly onto a telephone wire.

Mimic

Particles can be specified to follow a Layout motion path.

Those are most of the cool new features. There are more, like the ability to set a random death rate, setting particle size, and the option to turn off the preview screen for faster rendering of really complex setups. Some of the updates have not been mentioned because the crew at Dynamic Realities will be working on improving the features right up until the burn on the shipping CD. The new features give an already strong piece of software some really powerful new abilities. Keep in mind that there are other programs, like Steamer and Genesis, that are starting to use the Particle Storm architecture. With these new functions and finer controls, LightWave is sure to break some new ground in the field of particle animation.

So here you have it: two powerhouse programs for your animation delights. I am looking forward to using them in my daily work, as well as seeing what animators out there can come up with.



Property	Value
Name	Ground Plane
Position	
X	-50.0 m
Y	-5.0 m
Z	-50.0 m
Orientation	
H	0.0 deg
P	90.0 deg
B	0.0 deg
Scale	(100:5:100)
Elasticity	67 %
Roughness	50 %
Avoidance	
Time	0 sec

The new interface for Particle Storm 2.0, showing the tree control. OpenGL shading is also available for previewing collisions.

This "grateful-ly dead" rocker's hair was created using Particle Storm 2.0's Strands feature. An AVI is on the cover CD.

EAR TO THE WALL

Seamless Textures CD makes texturing simple

Seamless
Textures
you can Really use
MARLIN STUDIOS

Graphics studio and game developer Marlin Studios recently announced the release of its much anticipated "Seamless Textures You Can REALLY Use" CD. The CD features over 250 seamless textures, most with matching bump maps, for use in a variety of computer graphics applications, such as 2D/3D art, computer games, web content development, multimedia, film industry CGI, and many other applications. The collection, created almost completely from original photography, includes hard-to-find, photorealistic real-world textures, such as peeling paint, rust, corrosion, and much more.

The "Seamless Textures You Can REALLY Use" collection is divided into 27 categories, making selection quick and efficient. Thumbnail images are included with each category, and each texture has a memorable name, such as "Chinese Cafe Blue," "Rusty Swirl," or "Dripping Orange Rust.". While the CD includes many common categories like Bricks, Marbles and Woods, there are more exotic categories like Rusty Surfaces, Paint Surfaces and Wickers.

The Marlin Studios "Seamless Textures You Can REALLY Use" CD sells for $99 plus s/h. The CD can be ordered from the Marlin Studios web site, where users can also download free texture samples and view thumbnail pages. The CD can also be ordered by phone from Marlin Studios' distributor, Tachyon Technology Corp. in Seattle, Washington, at 206-622-7805.

http://www.marlinstudios.com Email: tmarlin@marlinstudios.com

New LightWave plug-ins

Motion Designer: Motion Designer is a displacement plug-in for LightWave. It transforms the polygons in an object based on an elastic body model to simulate the effects of wind, gravity or object movement. Motion Designer creates a special motion file, which is then loaded into the scene (similar to Particle Storm).

For more information, visit http://www2m.meshnet.or.jp/~inodx/md/indexe.htm.

ShaderMan 2.0 now available: Dynamic Realities announced recently that ShaderMan version 2.0 is now available. ShaderMan is a collection of 25 surfaces based on RenderMan algorithms. In version 2.0, you will be able to envelope and animate most of the surfaces, as well as preview the surface without rendering. ShaderMan 2.0 is available for Intel, DEC Alpha, Macintosh, Sun, and SGI. Users of version 1.0 can upgrade for $75. Dynamic Realities can be contacted at 800-944-5520, or visit their website at www.dynamic-realities.com for more information.

FIBER FACTORY 2

Fiber Factory 2 ships! Metrografx has started shipping Fiber Factory 2! New to this version is the addition of a real time, OpenGL interface. Gravity amount, length, and width of each fiber are completely controllable. Special effects such as random jitter, contour, curl and kink provide realism. Three additional Layout plug-ins are included with Fiber Factory 2- *Anchor* creates fibers that react to gravity, inertia, and wind. Use bones in combination with *Anchor* to style and animate the hair. *Fshade* gives specular, diffuse, and ambient lightning effects. This means create more hairs using less memory! The third plug-in is *Fflt* pixel filter, which can create dense hair complete with lighting and texture mapping.

Fiber Factory 2 is selling for $199.95 on the Intel, Mac, and Alpha platforms, with the upgrades costing $80.00. Free demos are downloadable and more information is available at www.metrografx.com.

More New LightWave Plug-ins

Random Point Distribution Plug-in This plug-in from Pontari Productions will, in one step, make a random number of single point polygons. This can be extremely handy for making star fields. It also makes smoke/dust blobs for use with Super Glow product, as well as Steamer that comes with LightWave 5.5.

http://www.pontari.com/lightwave
Matthew Sorrels matthews@pontari.com

New Fractal Noise Plug-in! Fractalizer, a new fractal noise plug-in from Dynamic Realities, is now available. Surfaces like rust, water, plasma, dirty objects, volumetric fog, and more are easily achievable. The fractal morphing feature makes animating the surfaces simple, for results like a burned out hole or electric sparks. With a 5 color spectrum, almost infinite colors are possible. Fractalizer is available for Intel, DEC Alpha, Macintosh, Sun, and SGI.

Contact Dynamic Realities at 800-944-5520 to order, or visit www.dynamic-realities.com for further information.

Image created using Fractalizer

Polygon Reduction qemLOSS (Lightwave Object Surface Simplification) is a free plug-in available for the SGI, Intel, and Alpha platforms. It is a Modeler plug-in that uses a surface simplification algorithm in an attempt to reduce the number of polygons in a LightWave object. The plug-in allows the user to rapidly produce good quality approximations of excessively detailed polygonal models.

This plug-in does polygon reduction similar to Decimate, but uses a completely different algorithm. It is geared toward objects that will be used for level of detail type applications and objects that can be used as low-res stand-ins while doing scene layout. Decimate was intended for applications where the object's topolog needs to be preserved (medical 3D isosurface reconstructions). qemLOSS does not have the restriction of preserving the topology, and therefore in most cases, will provide more reduction but still provide a good approximation to the original object.

More about this plug-in and its usage can be found at
http://amber.rc.arizona.edu/lw/qemloss.html

LumeTools is coming! The SoftImage version of this plug-in was used to create the superior surfacing effects in *Riven*, the sequel to the popular game *Myst*. LumeTools will be available as an entire collection of shaders, or in sets of related shaders. The sets are:

LumeLandscape - Includes *Landscape* for mapping textures to terrain models using slope, height, and noise functions, *Mist* for creating truly layered fog with depth fading, and *Façade*, which places 2D cardboard cutouts in the scene and rotates them with the camera. A great memory and time saver!

LumeWater - Comes with *Ocean* to simulate animateable and looping waves on large water bodies, *Water* for modeling the physics of the reflective, transparency, and coloration properties of water, as well as letting objects know when and where they are wet and applying textures accordingly, *Submerge* for underwater depth fading effects, *Wet* to facilitate the creation of wet objects with saturated colors and specular highlights, and *Stain* for confining textures with the boundaries of other objects.

LumeLight - Includes *Glare* to provide realistic glares and flares from any bright source, not just lights, and models the way the human eye scatters light. *Translucency* creates truly translucent textures with blurred transparency and backlighting, scattering effects. *Glow* provides internal lighting to an object; *Night* simulates the eye's rods and cones, making colors appear desaturated in dim lighting (i.e. moonlight); *Beam* gives a basic atmospheric effect around lights without shadow calculations for high-speed renderings, and *Illumination* provides a physically accurate lighting model for light sources.

LumeMatter - Includes *Metal* and *Glass* for modeling the reflective and transparent properties of these surfaces, including lighting shifts near edges, reflection blurring, coloration, and more, and *Edge* for special edging effects like a fuzzy mouse, a notched peg, a blurry ghost, or a jagged rock.

LumeWorkbench - Provides simple yet important tools, like *WrapAround* for a 360 degree rendering, *Distortion* for simulation wide-angle lenses, *BumpCapture* to create bumpmaps right in LightWave, and *Adjustments* for adjusting the coloration, brightness, contrast, hue, etc. of images. Can be applied on a per-object basis.

Contact Dynamic Realities at 800-944-5520 or www.dynamic-realities.com for more information.

Image created using LumeWater

NewTek announces CEO change

NewTek, manufacturer of industry-leading 3D animation and video production products, announced on March 2, 1998, that Tim Jenison, the company's founder and chairman, will assume the position of interim President and CEO, succeeding Dwight Parscale. Mr. Parscale will be leaving to pursue other opportunities.

Tim Jenison is considered the visionary force behind the desktop video revolution. Since founding NewTek in 1985, Jenison has led in the design of a series of ground-breaking products and has guided the company as it continues to pioneer innovative and affordable tools for computer-generated animation, video , and film special effects.

"NewTek creates products that not only reflect the company's entrepreneurial spirit, but that also empower users to become entrepreneurs themselves," said Jenison. " In 1997, NewTek achieved its highest sales year ever with a 46% growth rate for the year. I am excited about continuing with the development of future ground breaking products and guiding the company to even greater success in 1998," Jenison continued.

NewTek will begin an executive search for a new president and CEO immediately. The company will seek a technology executive with established industry knowledge and demonstrated experience in high growth companies.

About NewTek
With headquarters in San Antonio, NewTek is the leader in providing full-featured video editing and special-effects tools that allow anyone to produce professional video and graphics. The company's products are used worldwide on projects from home video to feature film, including some of the most widely anticipated recent blockbusters, such as *Men In Black, Titanic, The 5th Element* and *The Jackal.*

Contact NewTek at 8200 IH-10 West, Suite 900, San Antonio, TX 78230 http://www.newtek.com.

Upcoming Events

NewMedia98 Multimedia computer show in Toronto, Ontario, Canada. Conference dates May 12-15. Exhibition dates May 13-15. Exhibitors include NewTek (booth 613), Dynamic Realities (booth 718), Adobe Systems, Inc. (booth 1605), DPS (booth 423), Kinetix Autodesk (booth 1107), Avid Technology, Inc. (booth 813), and MetaCreations (booth 1411). Conference highlights include Creative Deal Making and Financing Strategies, Application of Motion Capture for Virtual Actors and Character Animation, and Photoshop classes. One of the feature exhibits is the Mandala Virtual Reality System by The Vivid Group.

For more show information, visit http://www.newmedia.ca, or call 905-660-2491

LENY (LightWave Expo New York) This LightWave focused show takes place Saturday, May 23rd in the Big Apple. LENY 98 will feature a bigger lineup than ever before. This year's seminars will feature Ace Miles, Lee and Ken Stranahan, Matt McDonald of VSI and Eric Barba of Digital Domain.

Register now at www.msd-nyc.com, or call 888-MSD-NYC1 (673-6921) Admission to the exhibit area is free, and registered attendees have a chance to win a free copy of LightWave 3D 5.5! Don't miss LENY '98!

ON TOPIC

As I sit here and write this column, yet another report is on TV describing the latest box office totals for James Cameron's motion picture, *Titanic*. The movie is well on its way to becoming the most successful film in history. The movie is quite good and very entertaining, but more importantly to the LightWave community, it contains some of the most impressive and realistic computer graphics seen on the big screen thus far.

While it's exciting to be able to say that LightWave played a major role in the creation of Titanic's special effects, it must be noted that LightWave was just one of many different tools used to produce the film. This brings me to the subject of my little diatribe for this issue.

Far too many LightWave animators seem to live in a glass bubble where they think that everything can and should be done exclusively with LightWave. Granted, it is possible to do some amazing things with just this one tool, but why would you want to limit yourself this way? You never hear of a carpenter trying to build an entire house with just a screwdriver, and the reason for this is that it just doesn't make any sense! So yes, LightWave was used in the production of Titanic, but it would have been impossible for the folks at Digital Domain to do all of the work without the aid of a number of other graphics tools.

Many times a person will ask the LightWave mailing list for suggestions on how to create a certain effect, and when the answer involves the use of some sort of commercial plug-in or program, the person will get upset that there is no easy way to create this effect without purchasing additional software. Well folks, it is time to accept the fact that LightWave is NOT a Swiss-army-knife for video production. It is a 3D animation program, and a very good, very affordable one at that, but if you want to get "serious" about doing 3D animation, sooner or later you are going to have to bite the bullet and add some other tools to your toolbox.

If you're on a tight budget, like most of us that are working out of our bedrooms and basements, then you probably cringe every time you hear the word "purchase." But you also have to consider the amount of time and hard work you will save yourself by adding a few commercial plug-ins or programs to compliment LightWave. Many plug-ins, like Surface Effectors, Lock & Key, and FXtremePro, will eliminate many hours of tedious work in setting up your LightWave scenes, while a paint program like PhotoShop, or compositing software like After Effects, or video editing applications like Premiere and Video Action Pro, will give you the tools you need to put your production together quickly and with professional results. No, these programs are not free, and most of them are not even cheap, but many of the tasks that they perform are incredibly time saving or just impossible to do without.

The point here is that there are times when you are not going to have any choice but to use other programs in conjunction with LightWave, and you must realize that this is NOT a bad thing. Learn what each program does best, learn how to integrate the tools together efficiently and, with a lot of practice and a little luck, in no time you will be producing your own billion-dollar blockbuster.

Here is this issue's selection of questions and answers from the LightWave mailing list...

Q I've modeled a complex MetaNURB object with LightWave 5.5, and it looks great in Modeler's OpenGL preview. However, it is completely invisible in Layout. The object renders correctly, but I just can't see it in Layout. What is going on here?

A This will happen if your model has any unattached points in it. To check for unattached points and remove them, do the following:

1. Load the object into Modeler and switch to Point selection mode.

2. Press the 'w' key to bring up the Point Statistics window.

3. If you see a number greater than 0 next to the "0 Polygons" statistic then your object contains points that

are not attached to any polygons, and this will cause display problems in Layout.

4. Click on the "+" button to select these unattached points and then Cut them from your object.

5. Resave the object, load it back into Layout, and it should now be visible.

Q I don't like the new Surface Color requester that is used in LightWave 5.5. Is there any way to get the "old-style" Surface Color requester back?

A Yes! If you hold down the Shift key while clicking on the Surface Color button, the old requester will appear (the one with the RGB slider bars), and if you aren't holding down the Shift key, the new requester will appear. This "trick" only works in Layout's Surfaces panel, and is not available from Modeler's Surface menu.

Q I need to render a large animation for output to my PVR, but don't have enough time to get it done. Is there any way to decrease the render time without losing too much image quality?

A For PVR output, you would normally render your animation at the D1 setting with a Custom Size of 720x480 pixels, but you can save a good bit of render time by switching to the Square Pixels setting and rendering at a resolution of 640x480. Then, when you send the images to the PVR, make sure that the Auto-Size option is turned on, and the PVR software will automatically stretch the width of the rendered images to the correct size of 720 pixels with little or no noticeable decrease in the quality of the final output.

If you don't have enough time to render the animation at the 640x480 resolution, then another way is to render the animation in two passes, first as a low-resolution image sequence, and then the second pass will be a resize operation that uses LightWave's powerful Anti-Aliasing feature to increase the resolution of the rendered image sequence and eliminate any jagged edges that would normally result from increasing the size of an image. Here's how to do this:

1. Go to the Camera panel and change your output settings to Low-Res D1 with a custom size of 360x240 pixels.

2. Render your animation as an image sequence. This will create a series of images that are 1/4 the image size needed by the PVR.

3. Go into Modeler and create a simple flat plane in the Front view. Its dimensions should match the size of the desired PVR image size, so a plane that is 720 meters by 480 meters will work fine. You can also use any variant of this size, like 7.2 meters by 4.8 meters.

4. Go to the Objects panel and load the flat plane into Layout.

5. Go to the Images panel and load the image sequence that you rendered in step #2.

6. Go to the Camera panel and change your render settings to the required PVR resolution, which is D1 with a Custom Size of 720x480 pixels.

7. While you're in the Camera panel, be sure to set the Anti-Aliasing preferences to something that will work efficiently. In most cases, you can use Low or Medium Anti-Aliasing with an Adaptive Sampling Threshold of 50 or more, but it may require a few test renders to get the Adaptive Sampling Threshold set just right.

8. Go to the Surfaces panel and apply the image sequence you loaded in step #5 as a Planar Color Map. Click on Automatic Sizing with the alignment set to the Z axis to make sure the image sequence is sized and positioned to fit correctly on the flat plane.

9. Raise the Luminosity level to 100% and lower the Diffuse level to 0%. Make sure that all of the other surface attributes are set to 0% or their default settings.

10. In Layout, switch to the Camera view, and position the flat plane object so that its outer edges fit just inside the rectangular dotted lines that mark the boundaries of what the camera will see when the image is rendered.

11. Before rendering out the animation, you will need to make sure that the Anti-Aliasing Adaptive Sampling Threshold is set properly (as mentioned in step #7). To do this, turn on the Show Render In Progress option and turn off the Automatic Frame Advance option. Now test render a few frames to make sure that Anti-Aliasing is being applied to enough of the image without slowing down the render too much. If you have difficulty getting Anti-Aliasing to work the way you need it to, then try switching to one of the Enhanced Anti-Aliasing modes available in the Camera panel. These modes generally do a much better job of smoothing out jagged edges in an image but they will increase the time it takes to render each image.

12. Once you are satisfied with the results of your test renderings, turn off the Show Render In Progress option, turn on the Automatic Frame Advance option, render out your animation, and record these resized frames to the PVR.

You should note that using LightWave's Anti-Aliasing feature in this way will eliminate many of the hard edges in your resized animation, but the final output may have a very soft, fuzzy-edged feel to it, sort of like applying the Camera's Soft Filter option. There is no real way to avoid this, but it will be less noticeable if you get the Anti-Aliasing Adaptive Sampling Threshold set "just right". Remember that with this technique, you are definitely sacrificing image quality for shorter render times. If you need to have sharp, clean edges in your final output, there is no substitute for rendering the image at the correct final output resolution.

Q Will Field-Rendering make my animation look better? And if so, how much longer will it take to render?

A In order for you to understand how Field-Rendering will help your animation, you need to know that all video display devices work in an interlaced mode. Essentially, this is a fancy way of saying that your TV does not display 30 frames per second, but it does display 60 half-frames per second. One half frame will update all of the odd-numbered horizontal lines on your TV and the next half frame will update all of the even-numbered horizontal lines. Field-Rendering will allow you to have your animation play back much more smoothly because the action will be updated 60 times per second, as opposed to the 30 times per second you get with normal frame rendering.

This works beautifully for animations that have a lot of movement in them, especially if the camera is moving. So the answer to your first question is a definite yes, Field-Rendering will make your animation look much better when it is being played back. However, it should be noted that if you pause the playback of a field rendered animation, you may notice that the image is blurry or has an undesirable flickering effect. This is normal, and it occurs because your animation will always consist of two fields, and the odd/even numbered lines in each field can sometimes be very different from each other. Normal frame rendering will always give you a nice, sharp image to pause onscreen. This is because both fields in the frame are the same and always match correctly, so Frame Rendering will give you sharper paused image and Field Rendering will give you smoother playback.

As for an increase in render time, you will notice that Field Rendering does take a little bit longer to render, but it is usually a very small increase, and the improved quality of your animation playback is well worth it. One thing to keep in mind when using the Field Render mode is that even though you are rendering the animation's fields separately, you are not creating any extra image files. So if you were rendering a 200 frame animation, the first rendered image file would contain a field from Frame 1 and a field from what could be considered Frame 1.5, but in the end you will still have 200 image files to record to your video output device. Also, most video output devices have a special Field Playback option you must select in order for your Field Rendered animation to play back correctly. Check the manual of your video output device for more information on Field Playback.

Q Is there an easy way to convert a Spline curve into a string of 2-point polygons?

A Yes, all you have to do is copy the first point in your Spline curve to another layer, turn it into a polygon, put the original Spline curve in the background layer, and then use the Rail Extrude tool to extrude the 1-point polygon along the Spline curve's path.

Q I need to create an image that will be output by a local printer. What are some of the things I need to know about creating print-quality images?

A Ideally, this question would be answered with a full-length tutorial, but here are a few things to keep in mind when attempting to create professional print-quality images:

1. The image should be rendered at no less than 300 dpi (dots per inch). This means that if you wanted to create an image that would fit on a 5"x3" business card, you would need to set LightWave's Camera preferences to Print Resolution with a Custom Size resolution of 1500x900.

2. The image's dpi information must be saved with the image in a format that your printer can correctly read. This means that you are going to need a good image manipulation program, like Photoshop, which will allow you to manually set the number of dpi for the image, and then have that information save correctly in an image format like JPEG or TIF. (JPEG is preferable because even with the Maximum or High quality settings, the image file will usually fit on a 1.4mb diskette).

3. The image's color information needs to be changed to the CMYK format, which is the 4-color print standard for maintaining high-quality output. This is another operation that cannot be done with LightWave alone, but a program like Photoshop can handle the color mode change from RGB to CMYK quite easily.

4. Most importantly, be sure to ask questions! Get on the phone with the printer that you plan to use and find out as much as possible before walking into their facility. If they are unwilling to answer your questions, then find another printer. Some places charge a hefty fee for their printing services, and it is important that you avoid making costly mistakes, especially if you are doing work for a paying client. In this case, I recommend you do a few test images with varying color ranges, just to get a clear idea of what to expect from the printer you will be using...when it comes time to hand your client's work over to the printer, it is much better to have as few surprises as possible!

Q I'm having a problem with MetaMation, where my MetaForm_Target object seems to lag behind the MetaForm_Shaper object by a frame or two. How can I fix this?

A This usually happens if you load the high-polygon "target" object before you load the low-polygon "shaper" object. It is important to load the low-polygon version before loading the high-polygon version, and this is also true when you are actually applying the two different MetaMation plug-ins. Be sure to apply MetaForm_Shaper to the low-polygon "shaper" object before you apply MetaForm_Target to the high-polygon "target" object.

That is more than enough for this issue...see you again next time for more On-Topic discussions from the LightWave mailing list!

To submit questions for Dave Warner, please email OnTopic@keyframemag.com, or fax to 805-264-4998.

AUTHOR PROFILE

In addition to running the Internet's LightWave mailing list, David Warner produces the TutorVision 3-D CD series of interactive, multi-media CD-ROMs for LightWave users. For more information about the LightWave mailing list, an archive of Frequently Asked Questions from the mailing list, and product information for TutorVision3D, please visit the Event Horizon Studios web site at http://www.tv3d.com or send "snail-mail" to David Warner at: P. O. Box 310 - New Baltimore, NY 12124.

• LEARNING TOOLS & INDUSTRY LEADERS ON THE WEB

WAVE SURFING

by Anupam Das

A few years ago, the Internet would not have been nearly as valuable of a resource for LightWave animators as it is today. The 'net plays an integral role in businesses worldwide, and it comes as no surprise that LightWave 3D has its own virtual existence in cyberspace. From special interest newsgroups to web-sites featuring animators and visual FX houses, the number of resources for the beginner and expert alike are mind boggling.

That's great, but as we all know there are good web-sites that provide valuable information, and there are those that leave much to be desired. Many people don't have time to "surf the net" anymore. This article's main purpose is to do exactly that for you. We will highlight some of the LightWave related web sites that showcase the work of the top 3D artists and visual FX houses using LightWave, as well as those which serve as a technical resource for LightWave users.

The Sites

Our first stop is **www.stationxstudios.com**. Yes, this is the official domain for Station X Studios. As many of you know by now, Station X Studios was formerly PlanetX, the team that delivered ground breaking visual effects for *Titanic* at Digital Domain. Now they have their own ship to sail, and from the looks of their website, they plan to do it in a spectacular way. The website is one of the few that really highlights the use of LightWave in *Titanic*, as well as who was responsible for what shot, something I am sure we are all curious to find out. While there, make sure to visit Aristo Menis Tsirbas' (FX Supervisor) homepage.

The LightWave Mailing List serves as a great resource. Here you can read about the latest developments relating to LightWave, as well as receive direct support from the industry experts themselves. Big names in the LightWave community frequent the mailing list, so this is your chance to get to know them. This mailing list also has a great webpage full of helpful resources. Just link to http://www.albany.net/~dwarner/LW.html. Dave Warner has been kind enough to keep an archive of previous months, as well as having a great search engine to help you along. I have found this to be an indispensable source of information. If you are interested in developing plug-ins for LightWave 3D but don't know what direction to pursue, just talk to the other plug-in developers on the Plug-in Mailing List. Don't forget to view the Mailing List contest page that showcases some spectacular LightWave renderings.

For those who do not think spectacular landscapes can be created with LightWave, think again! Point your browser to http://www.bestweb.net/~wirewolf. This is the homepage of Eric Knight Holbrook. His gallery is a must-see. Eric has a very good sense of space, color and mood, but the images speak for themselves.

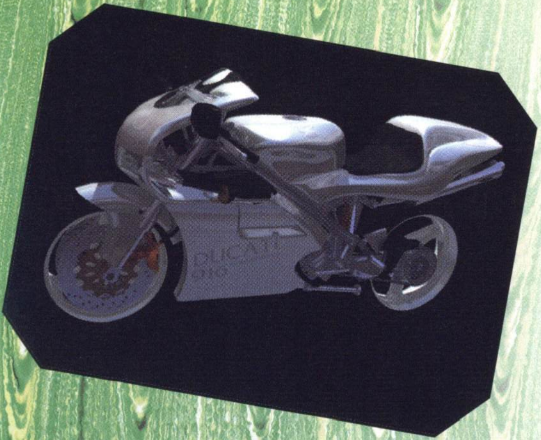
Unfortunately, that is all we have time for this issue. If you find a resource on the Internet that is especially helpful, or a page that shows off LightWave or Inspire, let us know so we can pass the word along! Until next time, keep surfin'.

FireCD
Pyrotechnics Footage on CD-ROM
ONLY $49.95
To Order Call Exodus Multimedia
888-3EXODUS (339-6387)
http://www.firecd.com

Reader Service #109

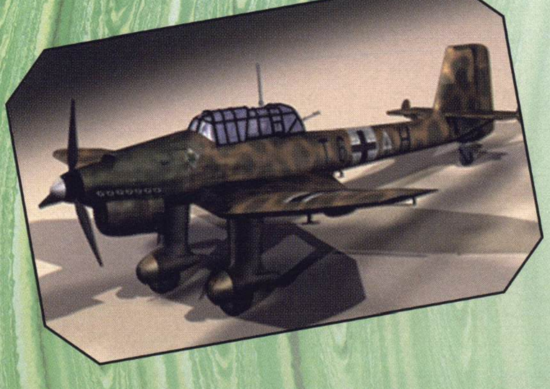
Authorized NewTek Center For Canada
VIDEOLINK
ANIMATION & DIGITAL VIDEO
NewTek
Authorized Amiga Repair Shop
Your LightWave 3D one-stop shop for
CD ROMS, Plugins, Videos, Books, Training,
LightWavin' Magazine, 600 MHz Digital Workstations & more!
LightWave 3D for Amiga, Dec Alpha, Macintosh, Intel & More!
LightWave 3D/Oxygen Bundles Priced To Sell!
Everything You Need Always In Stock!
53 Lucy Ave. Toronto Ontario
M1L1A1 Canada
Call (416) 690-1690 Or 1(800) 567-8481
www.videolink.ca
powered by AMIGA
DPS
AMIGA, NEWTEK, DPS, YC+, Digital, Oxygen, Sanyo

Reader Service #108



Learning Curve

The Digital Classroom for LightWave animators...



Issue 6 Topics

Perfectly Imperfect Surfaces
5 Minute Lightning Bolt
Modeling a Complex Object
Character Shop

PERFECTLY IMPERFECT SURFACES

by William Capozzi

As we all know, one of the attributes of Computer Generated Imagery is perfection. Compared to the real world, which is quite grimy and dirty even in the most well-kept places, computer renderings almost always contain clean surfaces, sharp edges, and no signs of actual usage.

In a cartoon world, cleanliness may be desirable for a given effect, but let's say you need to create an element that requires a little more realism. Our goal in this tutorial is to look at a few ways to do some inexpensive damage to your 3D objects.

For the benefit of producing similar results, I should mention that I am working on an Intel Pentium 150 running Windows NT 3.51, with LightWave 3D 5.0 installed, using only the plug-ins that were supplied with that version. Also for this tutorial, you will need some sort of image painting and processing program. I am using Photoshop 3.0.

Start by making three images in your paint program. The first a dirt map (Fig 1A), the second a bump map (Fig 1B), and the third a variation on the second for the specularity map (Fig 1C). Because the three image-maps are quite simple, I will not go into details on the creation of them. If you are not familiar with how to paint such an image, try using soft edge brushes and use mainly black and white.

Now that we have some images, we need something to put the images on. We will use a vase. It allows for easy image placement and quick setup. Run Modeler, and using the Sketch tool, draw a shape that looks like this (Fig.2) in the front view. Remove any extraneous points for a smooth curve, then select the bottom points and use Set Value to set the points at 0 on the Y axis. Select the two end points and use Set Value again to set the points to 0 on the X axis. Using the Lathe tool, lathe your object 360° with 32 sides on the Y axis to look like this (Fig.3). Voila, a vase.

Next, check to make sure the polygons are facing out. Then press 'q' on the keyboard and give it a surface name of 'vase'. While we are here, set any medium to light color you want to use, set Specular to 10% medium, and click on Smoothing. Save your object out as 'vase.lwo'.

Start up Layout and load in the vase object and the three images previously created. Open the Surfaces menu and click on the Diffuse 'T' button. Use a cylindrical map on the Y axis and select the dirt image

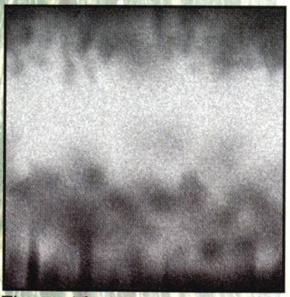


Figure 1A

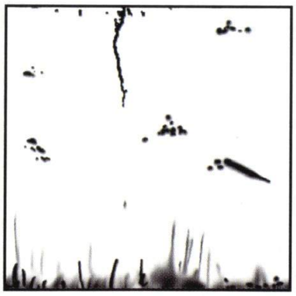


Figure 1B

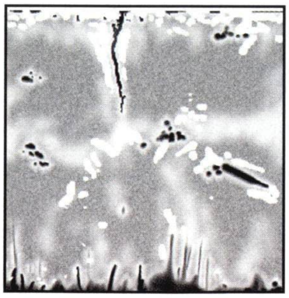


Figure 1C

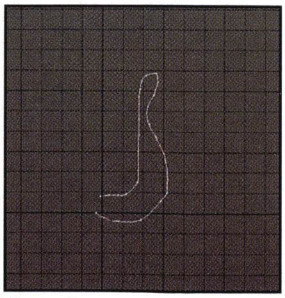


Figure 2

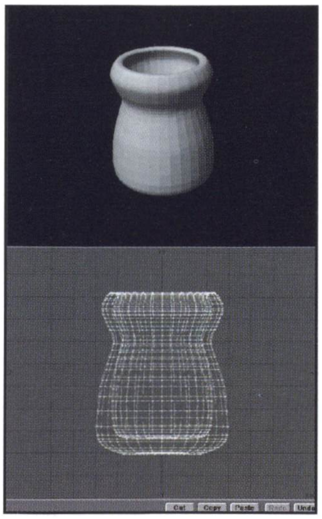
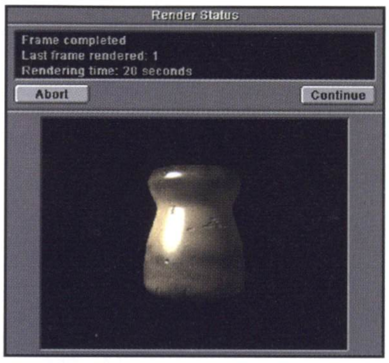


Figure 3



Frame completed
Last frame rendered: 1
Rendering time: 20 seconds

Figure 4

in the scroll button. Click Automatic Sizing and leave the rest of the values at their defaults for now. Apply the specularity image under the 'T' button for specularity, and the bump image under the 'T' button for bump map in the exact same way.

Make a quick test render of your object (Fig 4). From this point the work will be concentrated on tweaking the values, the lighting, and maybe some procedural textures to break things up a little more, depending on whether this is an ancient artifact or just something you forgot to dust.

A quick lighting scheme to use is this:

Ambient Intensity = 5
Light(1) = default values
Light(2) = point light
Intensity = 50%
Color R = 150, G = 180, B = 220
Location X = 2, Y = 0, Z = -2

This will add shadows and throw the light in different angles. Add a procedural texture to the bump map by selecting the 'T' button for bump map, selecting Crumple in the scroll button, and using the default values. This will add a roughness to the object. The rest of the values are up to you.

Experiment and become accustomed to the differences small and large changes can make. Suggestions are to change the texture opacities, low or high specularity, add another image to the surface, and add some reflectivity and trace reflections. (Figure 5 is my result).

Version 5.0 of LightWave 3D offers some great and simple surfacing tools. Once you know the basic procedure, the rest is just adding dimension, usage, and history to your creations. Layered surfaces give your objects a richness and realism that can make your work stand out, like the airplanes shown to the right. A single model aircraft is surfaced to look two different ways, using layered procedural textures and painted maps for each attribute.

Take advantage of LightWave's many tools and put the time in to learn how to create the digital images you and your clients want. The more adept you are, the easier the images will come.

AUTHOR PROFILE:

William P. Capozzi, Digital Artist, explores the many facets and planes of Computer Generated Imagery for film and video while operating Dedicated Digital, a 3D modeling and animation company. Inquiries can be directed to bcapozzi@mindspring.com.

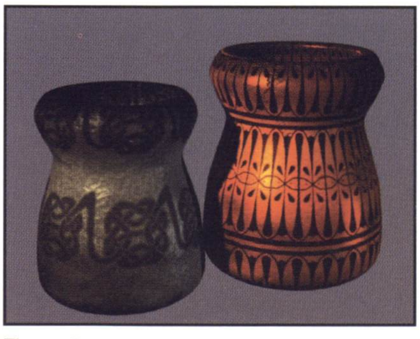
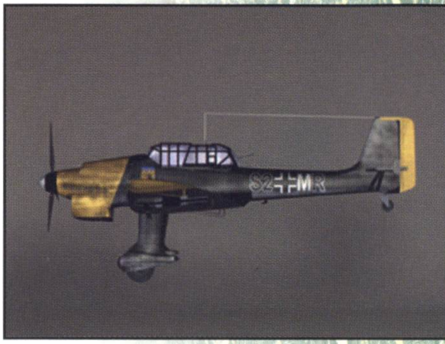


Figure 5

For more discussion of this topic, circle Reader Service #111



5 MINUTE LIGHTNING BOLT

by Patrik Beck

Here is the scenario: You have an opening graphic to a video about a new product, and your boss/client has been looking over your shoulder continuously for the last two weeks, confirming every change and tweak, so that it is perfect in their eyes. Then one of the directors takes a copy home the night before the big meeting, and comes back with the comment, "The opening needs to be more exciting."

Of course, they have no idea what they want or how to do it, there is no budget, and the meeting is that afternoon. When in doubt, add a lens flare, when in greater doubt, add lots of lens flares. Unfortunately, lens flares work best when they are on the screen for at least several seconds as they whoosh and bloom across the screen, and all you have time to render is a few frames to sneak into the opening to "make it more exciting." Not only that, but you used up your weekly allotment of lens flares on the secret star destroyer project you've been doing on the sly (what, you thought there was an inexhaustible supply of lens flares?)

Have no fear. There is a very popular graphic element that can easily be created and quickly rendered in LightWave that will be very effective when used on just a few frames: the ever popular and showy Bolt of Lightning.

The Popularity of Lightning

Lightning and electrical arcs have long been a staple of both film and print. Where would Dr. Frankenstein's lab have been without the constant buzz of the Jacob's Ladder arcing in the background? Images of lightning striking down from the heavens is almost as old as graphic representations themselves. The awesome discharge of electrical power together with the deafening crack of thunder is nature's original multimedia display.

Several paint and multimedia authoring packages offer 2D methods of compositing lightning-like effects over images. Both Image Effects and ImageMaster on the Amiga can do it, as can programs like Video Action on the PC. While the 2D compositing method might be acceptable for some things, there are times you need your electrical discharge to be true 3D.

The most obvious answer is to use VertiLectric, a plug-in from Blevins that does a wonderful job in creating lightning and electrical arcs, with many user adjustable parameters and options. But although you have been thinking about getting Vertilectric, you haven't quite gotten around to ordering it (and there is never any budget, remember?). Still, if you take that frame store with the title on it and zap it a couple of times with some lightning bolts, it is definitely going to add some juice to the opening. So what do you do? You roll your own lightning bolt!

Call Me Zeus

We will be going through the steps one at a time, but the basic concept of our lightning bolt is really pretty simple. We will take a connected chain of two point polygons and use displacement mapping to give it the distinctive lightning shape. Velocity added to the displacement map gives it animation, and some very basic use of bones allows us to control where the lightning strikes.

Do not worry if you have never used displacement mapping, bones, or falloff and velocity settings before. This tutorial uses them in very basic ways and should be a fairly painless introduction to these very useful tools. There will even be some suggestions on how to integrate these effects into your productions later on, so pay attention. Best of all, you should be able to whip this out in less then 5 minutes.

Creating the Geometry

Step 1: Create a single point polygon
Enter LightWave's Modeler and start with a clean screen. Go to the Polygon/Create menu and select "Points." Create a single point by centering the cross hairs at 0,0,0, and pressing the "Enter" key. Now hit the "p" key to change that single point into a polygon. (Congratulations, you have just created a one dimensional object.)

Step 2: Extrude the single point polygon
Go to the "Multiply" menu and select the "Extrude" function. We want to create an object that is one meter long with several segments, so use the numeric option by clicking on the button or by hitting the "n" key. Chose the "Z" axis. In the Segments box, enter the value of 100, and the Extent at 1 meter. Click on OK to close the numeric box. Once the box is closed, it is necessary to hit the enter key once more to execute the "make" command.

You should now have an object that starts at zero and shoots down the Z axis for one meter and consists of one hundred, two point polygons. Save this object as "Bolt.lwo" and return to Layout.

Adding the Displacement Map

Step 3: Load the object
In Layout, clear then scene, then load the "Bolt.lwo" object. (To help us understand what is happening, we will use the Top and

Side views in Layout to observe the effects of the displacement mapping before rotating the object for rendering in the Camera view.)

Step 4: Set the displacement
In the Objects panel, select the "T" for Displacement Map and use these settings. For Texture select “Fractal bumps.” For Texture Size set all the values to 10 mm. Change the Texture Amplitude to 0.05 and the number of frequencies to 1. Click on Use Texture to look at the effect the displacement map has had on the object. You should see the previously straight line distorted into an uneven jagged shape.

Step 5: Add velocity
Go back to the Objects menu and the Displacement Texture panel. To move the displacement texture down the length of the object, give the it a Z value of 10 mm. Because 10 mm is 1/100 the length of the entire object, the texture will travel the length over a hundred frames. To give the displacement a little more life, add a velocity of 1mm to both the X and the Y. Do a quick wire frame render to view the results of these values.

Step 6: Move center and add falloff
Right now the displacement is applied equally over the full object. By applying a falloff value of 100% on the Z axis, the displacement is at full strength at the center of the object and falls off to zero at the end. For positioning purposes, it is more useful to have it the other way around, where the pivot point has the minimum amount of displacement. Go to the Texture Center box of the fractal bumps displacement map for the bolt.lwo object. Set the Z value to 1 meter. This puts the center one meter away on the Z axis, and with 100% falloff, the displacement is completely gone by the time it reaches the pivot point. You may want to change the falloff to 90% or so, so that there is still a little bit of tickle left at the end.

Step 7: Position the camera
At this point we want to position the object for rendering. Go to the Camera view. Select the Bolt.lwo object and move and rotate it so that the pivot point is the center of the screen and the end disappears just off screen. You may want to load a background picture for target practice.

Step 8: Surfacing
Surfacing is pretty basic. Unless you renamed it on your own, the surface is probably called "Default", so change it to Bolt. Set the Luminosity to 100 % or more and Diffusion to zero so that it is not illuminated by any other lighting in the scene. Set the color to either a light blue or pale yellow, depending on your personal taste in electrical arcs.

Test render. You should see a pretty good approximation of a lighting bolt striking in the center of your screen. Remember that if you use the Glow function, it has to be selected in the Surfaces panel and also enabled in the Effects panel to work.

Step 9: Add bones for shape
Something that suggests itself almost immediately is to have the lightning bolt shoot over a logo. Position the Bolt.lwo at the beginning of the logo and keyframe it to move across the logo over a number of frames. This is one of the reasons that we reversed the direction of the falloff.

To give our bolt a more snake-like shape rather than a straight line, we will use three bones. For the benefit of those who have been frightened by bones as children, we will take it slowly.

1. Select the Bolt.lwo object.
2. With the Bolt.lwo object selected, click on the Bones button.
3. Hit the "+" button three times to give you three bones.
4. Click on "Rest Length" and set the rest length value for all three bones to 0.25 meters.
5. Keyframe bone (1) at 0,0,0.
6. Keyframe bone (2) at 0,0,0.5 m.
7. Keyframe bone (3) at 0,0, 1 m.
8. Be sure that you have keyframed both bone (2) and (3). Activate each bone by selecting it and hitting the "r" key. Do that for all three bones.

You can now position the beginning, middle and end of the lightning bolt like a slinky (Figure 1). Grab the middle bone and move and rotate it. Notice how the velocity of the displacement map still travels down the length of the bolt.

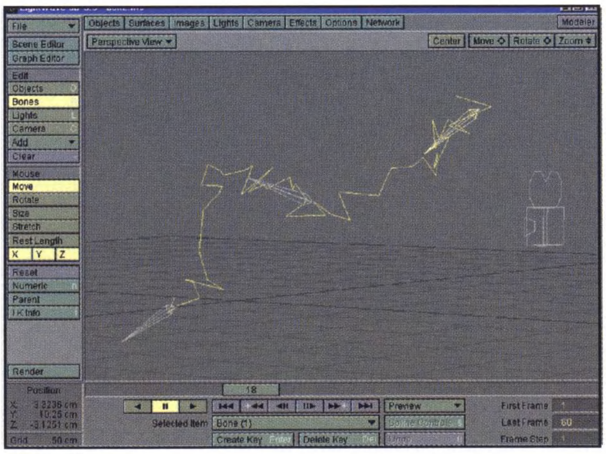
Step 10: Lens flare at tip.
If you have any lens flares left, parent one to the Bolt.lwo object. Because the displacement is almost completely fallen away at the pivot point, the lens flare should match the motion at that end of the lightning bolt. This will make it appear the area around the lightning strike is glowing. (It helps if you set up a real jerky envelope for lens flare intensity.)

Final Touches

An interesting effect is to turn on Motion Blur for the lightning bolts. Motion blur will render the stages of motion between the frames, giving the bolt an interesting, gauzy kind of look. If you want to use more then one bolt, simply clone the object and use slightly different values for the displacement map to give them different looks, along with choosing different line widths in the Particle/Line Size option box of the Objects menu.

Whether you are using this effect for a couple of quick jobs to liven up an opening, or using lightning as a transition to go from a "before image" to an "after image" as it crackles across the screen, lighting bolts and electrical arcs are a sure way to add some excitement to a graphic. While the method described here may not have all the bells and whistles of a dedicated lightning generator plug-in, by making slight adjustments to the suggested values you can achieve a lot of useful and creative effects. Best of all, you can do it all right within LightWave in just a few minutes. Ben Franklin would be proud.

For more discussion of this topic, circle Reader Service #112



Position
X: 3.3236 cm
Y: 10.25 cm
Z: -3.1251 cm
Grid 50 cm

18

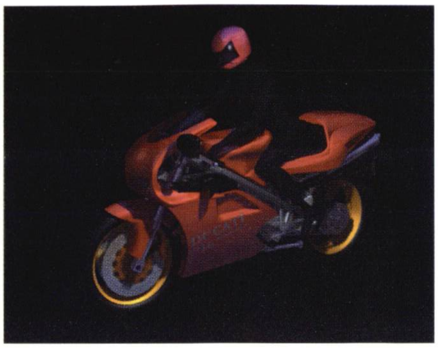
Selected Item Bone (1)

First Frame 1
Last Frame 60
Frame Step 1

Figure 1

Learning Curve

CD Content



MODELING A COMPLEX OBJECT

by James Hans

This project started from an idea I had in the Autumn of 1996. A friend and I were talking about our favorite motorbikes, ones we would like to own and ride. It turned out that our shared favorite was the Ducati 916. My friend knew that I was a keen modeler both in LightWave and with plastic kits. This conversation resulted in me going down to my local model shop and purchasing a Tamiya 1:12th scale model kit of a red 1993 Ducati.

The plan was to copy the model into LightWave as I built it, piece by piece, hopefully coming up with an extremely detailed model. By the industry standards of today, if you wanted to make an accurate representation of this kind of object you would just copy a small clay model with a digital laser scanner. That would have been easier, but in my opinion, when you are not on a deadline,e it removes all of the fun from the modeling process. You cannot sit back and be proud of something like that...yes, you put it into the scanner, but anyone can do that.

In the end it took me about 3 months to model this object. You may think that is a really long time, but I was a full-time student and had plenty of other things to do in my everyday life. There were times when I could spend several hours working on it and other times when I would not even touch my computer for several days. Obviously, this is not the most productive way to work, but this object was being modeled for fun and sometimes it is really important to get away from a project to gain a new perspective on it.

While following the instructions of the model kit, I started with the rear arm assembly...basically speaking, it is the arm that supports the coil springs and back wheel and is attached to the engine. This part on the real bike is a matte bronze colored metal. The only logical way of modeling this was to use five spline curves arranged in a stretched and twisting pentagonal shape. As each stage was completed, I would immediately go into Layout and texture the newly modeled part, just to see whether or not it looked believable from different angles and under different lighting situations. (Fig 1)

When recreating a complex object like this, you need a good eye to be able to judge widths and depths. I found this out the hard way when placing the back wheel into the rear arm assembly. The wheel ended up being wider than the space I left for it, even though both objects looked correct when viewed separately. Remodeling the rear arm assembly was not much of a problem, because I had already saved the spline curves used to create the original part. All I had to do was make a few minor curve adjustments and the wheel fit perfectly. By doing this part again I was even able improve the shape and allow for the grooves where the chain goes through. Saving temporary modeling elements, like the spline curves, is a very good thing to do, because it allows you quickly change and remodel the shape you are working on, while saving yourself a lot of grief.

Many of the connecting parts were created with slightly modified Constructive Solid Geometry primitives. The coil spring was a Torus rotated about 15 degrees, cut in half, and joined with another identical Torus to make a single coil-shape. I then cloned this a few times to make a full-length spring.

Next, I concentrated on the 916cc V4 engine. I like objects that are symmetrical because for most of the modeling process you only have to create one half of the object and then mirror it. I was able to do this for the cylinders, but the rest of the engine was much more tedious because many pipes and other pieces of metal joined the two sides of the V shape. For most of the engine, the shapes were just polygons drawn point by point and then extruded to make a solid shape. The pipes were created by extruding a simple disc along appropriately shaped spline curves. In my opinion, the spline tools are an extremely powerful part of Modeler; very intricate shapes can be made quite simply.

Now that the rear arm assembly and engine

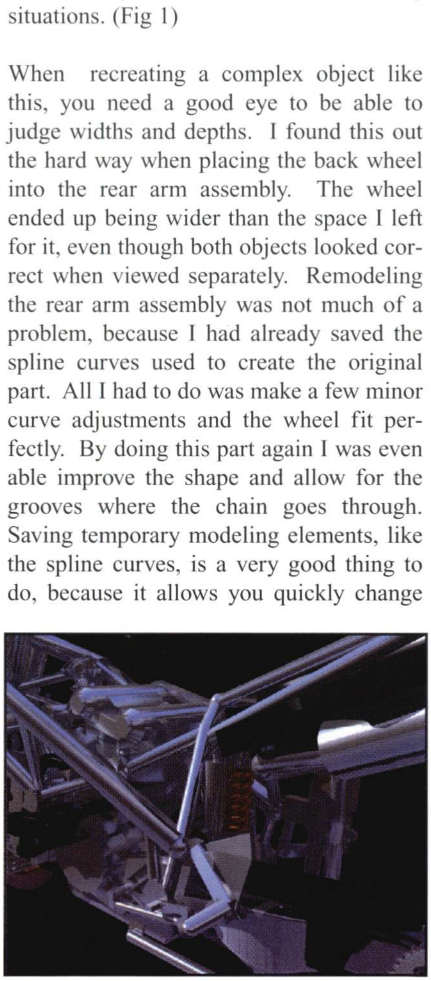


Figure 1

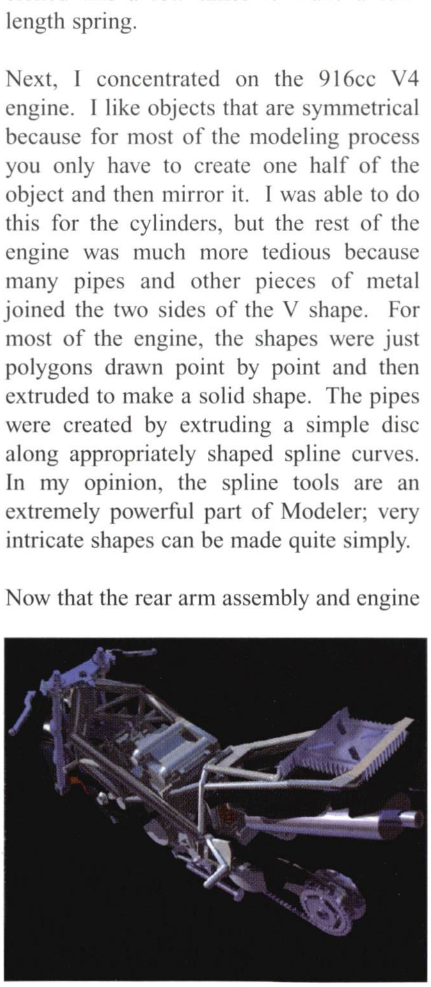


Figure 2

were completed, I could concentrate on the frame of the bike (Fig 2). In bike terms, the Ducati has one of the simplest frames because it is constructed entirely from tubes. This made it very easy to duplicate in Modeler. I ended up using the Cylinder an awful lot. Since the frame is almost identical on both sides, I was able to just create one side and then mirror it.

Now that I had finished creating the frame with the engine and rear arm assembly, I could add the exhaust pipes and silencer boxes below the seat. Not much thought was needed for these parts; the silencer boxes on the exhaust pipes are just cylinders.

With the bike really starting to take shape, I added the wheels and tires. (Fig 3) Looking at the model kit parts, I observed the cross section of the wheel and copied this simple shape by drawing with points only. I joined these points to make a polygon and then used the Lathe tool to extrude the shape around a center point. This gave me the bike's rim. I duplicated these steps to create the tire itself. The unfortunate downside to this was that the tire would look too slick when rendered. (I could have said that the model was the racing version of the bike!) A better and more realistic solution was to apply a bump map to the tire's surface. Using a bump map whenever possible will save memory and eliminate the need to model complex details, like the tire treads.

To duplicate the Pirelli Dragon tire tread, I simply sketched the tread pattern in Modeler, repeated the pattern, and then rendered out a large black and white image map. This image was then applied as a bump map to the rubber tire's surface.

The Ducati's tri-spoke rim was quite difficult to model. After a few failed attempts with spline curves, I ended up using MetaNurbs to create a smooth curve that was seamless in the middle of the rim and looked just right.

Attached to both of the wheels are the brake discs, which are essentially flattened cylinders with a lot of carefully placed holes. (Fig 4) The center of the front brake discs were constructed from a small cube with cylinders booleaned out of it. When the basic shape was finished, I beveled it several times to give a smooth, yet chunky look to it.

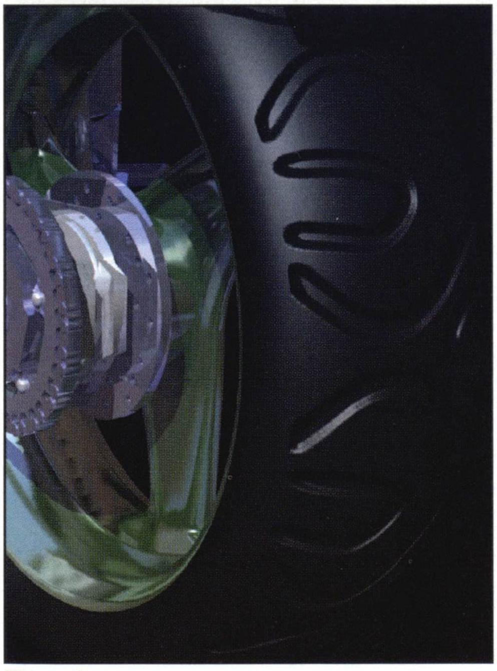


Figure 3

The chain segments on the bike were simple-shaped polygons extruded and then carved with the Boolean Subtract tool. This was then cloned along a path to follow the chain shape around the drive gear and into the engine.

Finally I got around to constructing the fairing and body parts. I started with the seat, which stretched back to the brake lights, so really it covered the entire back part of the bike. I tried to use spline patches to construct the seat, but it did not look very good. I ended up using MetaNurbs once again to create this part. You quickly learn which tools are best for each situation and here it was obvious that spline patches were not appropriate, although they were perfect for modeling the rear arm assembly.

Once I became familiar with MetaNurbs, (a tool that I had never used before starting this project) I managed to create the rest of the fairing very easily and all of the pieces fit together very well. The gas tank just set onto the frame and the final piece, known collectively as the headstock, worked out very well, fitting around the dials and handlebars.

The dials were scanned in from the original model kit's decals and then image mapped onto the object...simple but effective. I made many parts of the bike as simple as possible, but in the end the object turned out to have 160,000 polygons. A good example of this was when I constructed the radiator: I made the shape as simple as possible and then applied a checkered bump map to it. The bump map was only a 2x2 blurred black and white grid image map, but with a small texture size, a lot of depth, and just the right chrome texture, it made the radiator look like a highly detailed object. If I had not done this whenever possible, the polygon count would have been much higher.

Thinking back now, if I had used LightWave 4.0, which I bought back in November 1995 after seeing it at a show, many of the techniques that I used to create this object would not have been available and half of the bike would not have been completed. The Ducati object has had several uses as well as adding a nice piece to my demo reel. Its main use was in a road safety video where the bike (with a rider) was composited into some digitized road footage to demonstrate important safety tips to all people that ride motorcycles.

This may be a daunting task for the beginning modeler, but it is very rewarding as well as educational work.

AUTHOR PROFILE

James Hans an undergraduate in pursuit of his BA (Honors) degree in Computer Visualization & Animation at Bournemouth University in England.

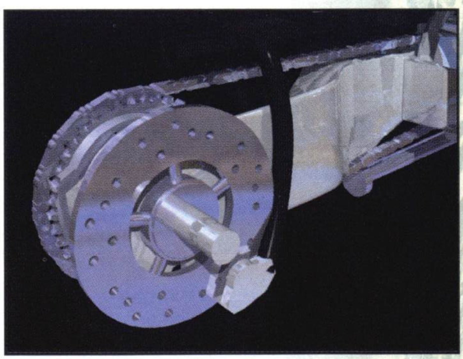


Figure 4

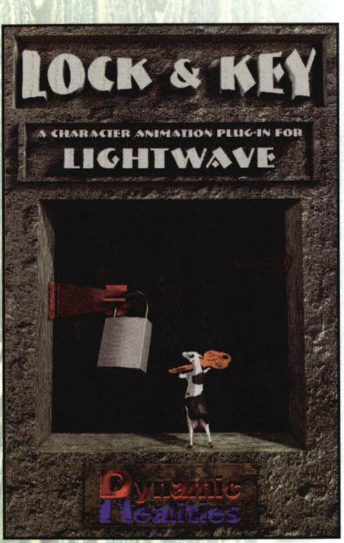
For more discussion of this topic, circle Reader Service #113

CHARACTER SHOP

by Doug Kelly

As promised in issue #5, this column begins a series introducing specific tools and techniques useful for character animation in LightWave. The first tool I recommend character animators add to their LightWave toolbox is Lock & Key, a plug-in developed by Meme-X and published by Dynamic Realities. This software is available from most LightWave dealers, or directly from the publisher at www.dynamic-realities.com.

One of LightWave's most egregious shortcomings is in the matter of constraints. Constraints, as the term is generally used in CGI, are methods of limiting the movement or orientation of an animateable item based on the movement or orientation of another item. In plain English, you animate one widget by using another widget as a handle. If a character's foot is constrained to the floor, you can pose the rest of the hierarchy any way you like, but the foot will stay in position. LightWave does have a number of built-in constraint tools. IK, for example, works pretty well as a look-at constraint. However, LightWave lacks an adequate position or orientation constraint. A position constraint nails down the item constrained by locking it to a second item, which I will refer to as the constraint. Position constraints are especially useful when you want a character to interact accurately with its environment: walking across a surface without skating; picking up and putting down items outside its hierarchy; even something as simple as opening a door by reaching for, grasping, turning, and pulling a doorknob.



An orientation constraint keeps the constrained item aligned with the constraint, like a bar magnet held over a compass needle. This is different from a look-at constraint. An item constrained to look at another item may assume any orientation necessary to point toward the constraint. However, if an orientation constraint is angled at 45 degrees, anything constrained to it will also assume an angle of 45 degrees, no matter where the constraint is located. This type of orientation is especially useful in constraining a character's hands or feet flat to a surface. It is also handy for tools or objects held by the character to be constrained to both the orientation and position of the character's hands.

How Lock & Key Works

Lock & Key provides both position and orientation constraints to LightWave. Constraints are referred to as Locks, and the constrained items as Keys. The set of a Lock with a Key is called a Combo. Once the Combo is set, the Key moves or rotates to match the Lock. This is easy enough to remember, if you think about it: a key belongs in a lock, and a key is a lot lighter and more moveable, so it makes sense that the Lock stays put and the Key is drawn to it.

Now for the tricky part. Lock & Key requires that a Key be the root of the hierarchy. This seems to require that the root of your character (generally near the center of gravity) would be constrained to the floor, or wherever the Lock was located. That does not seem very useful, does it? That is why Lock & Key also allows you to specify an Offset, a child somewhere in the character's hierarchy, that acts as a substitute Key for the root. For example, if a character's root was between its hips, and a Combo assigned it as a Key to a Lock located on the floor, the character would end up hip-deep in the floor. However, if you set the character's foot as the Offset, the foot would be constrained to the Lock, and the character would be standing on the floor. Once you understand these basics, setting up a hierarchy to use Lock & Key is straightforward, relatively simple, and tolerant of errors.

First, you should plan how you want your character to interact with its environment. Do you need it to walk on floors or other surfaces? How about leaning against a wall, or resting its hands palm-down on a table? Sitting in a chair, maybe? For any surface that you want to constrain to another surface, you need to place Offsets. If the character is composed of Parented objects, you will need to Parent nulls to the objects and position them on the surface you want to match the Lock. For a foot, you might set nulls at the bottom rear of the heel and under the ball. For a hand, the center of the palm generally works. If you want the character to sit properly, set an Offset… well, you get the idea. If the character is set up with Bones, you can set null Bones at the appropriate places in the hierarchy, positioned just as you would a null object Offset. The size of the Bone or null does not matter, the Combo is calculated from their centers.

Second, add Locks to your scene. You can use one Lock and simply animate it, or (for example) place a null for each footstep or handhold, and not animate the Locks at all. The approach you take depends on how you prefer to work and the demands of the action. Stay flexible, and experiment with different approaches.

Third, use Lock & Key to set Combos between your character's Keys (or Offsets) and the scene's Locks. If you want an orientation constraint, check the Lock Rotation box in the Combo Editor. Each time you set a new Combo or animate a Lock, you will not see your results until you make a keyframe, move to another frame, then return to the keyframe. This step is necessary to trigger LightWave's "evaluation", to retrieve the updated scene information from the Lock & Key plug-in. Do not forget to set the keyframe before the evaluation, or LightWave will forget your changes!

Once you have completed the Combo setup, you can animate your character using any other tools you like -- forward kinematics, inverse kinematics, other Motion plug-ins, you name it -- and the Keys will stay constrained to their Locks. You can pose your character doing anything from a simple walk to a hula dance, and the constrained feet will stay perfectly planted.

For a graphic example of the advantages of Lock & Key, take a look at Figures 1 and 2. Figure 1 shows the number of keyframes necessary to nail down foot slippage during a simple walk, using LightWave alone. Figure 2 shows the same scene, using Lock & Key to lock down the grounded foot. The motions of the two scenes were identical, or even a little in favor of the one using Lock & Key. Obviously, even simple Lock & Key setups can save a lot of work.

In my last column, I proposed a five-point list of criteria for an efficient character animation work strategy. Let's measure Lock & Key against those criteria.

Reduce Redundant Work

Absolutely yes, Lock & Key can save you a significant amount of time. You only have to set keyframes for the position or orientation of the Locks, then the start and end frames for the Combos. Without Lock & Key, you would have to set a key on nearly every frame, just to keep your character from sliding around. If you needed to modify any action between the foot and the root, you would have to delete and reset each of those keyframes. That is a lot of redundant work, and Lock & Key makes it unnecessary.

Support Incremental Detailing

Again, absolutely yes. With a basic Lock & Key setup controlling foot and hand placement, you can revise the character's skeleton to your heart's content (or your client's budget). This means you can start doing animation tests, animating the Locks as controls, with a crude Parented hierarchy of primitive objects. You can incrementally revise that character setup all the way to a seamless mesh animated using Bones or PuppetMaster. At each stage of revision, you have only to revise the Lock & Key Combos, assigning the new Keys to the old Locks. The essential motions of the character are contained in the Locks, not the Keys. This means better quality animation, since you continue to practice with and build on the animation of the character even while the character is being revised.

Provide Rapid Feedback

Lock & Key provides feedback in realtime, the same as native LightWave. The only proviso is Lock & Key's evaluation process. When you set a keyframe for a Lock, you need to step forward a frame and back again so LightWave retrieves the motion data from the Lock & Key plug-in. Unlike some other plug-ins, you do not have to render anything to see your results. Once the evaluation is done for a keyframe, you can scrub through the scene or make a Preview to see exactly the effect Lock & Key is having on your character.

Create Documents For Decision-Makers

This ties in to the idea of revisable character setups. Using Lock & Key, you can afford to do full animation tests with hierarchies built of crude bounding boxes, and use those tests to get client sign-off or production team critiques. This kind of iteration in the animation process is usually too expensive for anything below feature film production. With Lock & Key, you can afford to repeatedly tweak animation tests without blowing your budget.

Stay Flexible

Lock & Key frees you to experiment, to play with alternative character setups limited only by their root and twigs, without wasting a single keyframe of animation. As shown in last issue's column, the ability to adapt the animation of control handles to a completely new character setup gives you the ability to change the entire character, right up to the last minute. This is a huge boost to your flexibility, and can save your sanity, your budget and (possibly) your business.

Even given all these advantages, why do I recommend Lock & Key first, ahead of such useful plug-ins as PuppetMaster, PointAt, KeyPro or Con-Motion? The fact is that Lock & Key changes the design and workflow of even the most basic character setups. A setup intended for Lock & Key is essentially different from one designed for plain-vanilla LightWave, both in hierarchical structure and in the layout of control handles. A setup designed and optimized for standalone LightWave is not easy to adapt to work efficiently with Lock & Key, and vice versa.

If you are going to use Lock & Key at all, you should begin using it immediately, and become comfortable with and skilled in its use. Since Lock & Key is capable of constraining the control handles used for other plug-ins like PuppetMaster, you should consider Lock & Key a foundation plug-in. That is, other plug-ins can build on and enhance the setup that Lock & Key makes possible. On the other hand, if you design a character setup to use PuppetMaster, then try to modify that setup to use with Lock & Key, you will probably find yourself undoing much of your PuppetMaster setup and rebuilding it to take advantage of Lock & Key's labor-saving features.

Lock & Key makes any complex character setup easier and faster to animate, for both Parented object and Boned hierarchies. Better setups make learning and practicing animation easier, so I recommend Lock & Key for beginners. Lock & Key setups also save working professionals time and effort, so I recommend it for them as well. If you really enjoy setting a whole lot of redundant keyframes, then deleting and resetting them for every little change in the character's action, then you shouldn't get Lock & Key. If you actually enjoy animating, and want it to be easier so you can do more of it, Lock & Key is definitely a big step in the right direction.

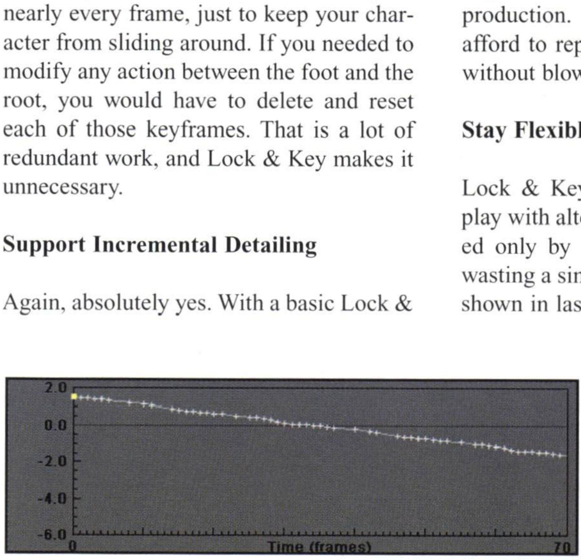


Figure 1

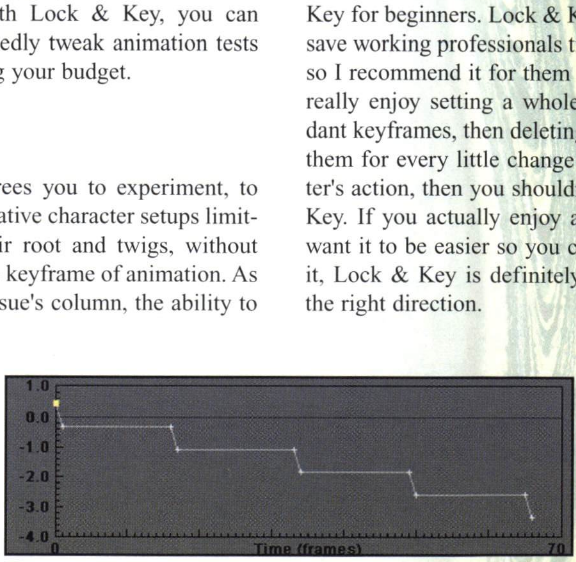


Figure 2

For more discussion of this topic, circle Reader Service #114

LightWave Mailing List

Opening Remarks
Patrik Beck

Reply
David Warner

Patrik Beck

If you are not currently a subscriber or have never heard of the LightWave Mailing List, listen up. It is the single greatest source the LightWave user has for information, inside tips, abuse, and frustration. It can also flood your mailbox with over one hundred messages a day, making you feel very popular.

The LightWave Mailing List is run by our own David Warner, and is accessed through your email. You subscribe to the list by sending Mr. Warner an email and your request to join. Once on the list, you will be inundated with email more or less on the topic of LightWave. The mailing list is like a big party full of loud, invisible people, and no snacks are served.

Being a lone animator and living in the Midwest, having any kind of dialog is a rare treat. An innocent post can take on a life of its own and generate a thread that veers wildly off topic. While this can be a welcome distraction if you are a really lonely person, it is frustrating if you need a solution to a specific problem and have deadline looming. Most questions that are posted get answered fairly quickly with various degrees of politeness. For many questions where the obvious answer is to look it up in the manuals, the person posting the question typically gets flamed as a software pirate.

One of the uses for the list is to track down reference material, information, and products that you may need to complete a specific job. You can also put out a call for specific objects or image maps.

A number of representatives of the 'big boys' have been known to check in at the mailing list now and then, and are usually pretty free about sharing trade secrets. More then one person has even found employment as a LightWave animator through it. If you can weather your way through the off-topic rants, the personal attacks, and threads that never die, you will find the LightWave Mailing List a valuable resource.

David Warner's Response:

About 10 years ago, a couple of friends and I would regularly load up the truck with beer and head to a diamond mine in Herkimer, New York, to do some prospecting and kill a few brain cells. Now these weren't "real" diamonds that we were looking for, but the work was hard and it was quite rewarding to find one of these little gems every once in a while. By the end of the day, we were usually pretty good and drunk, with only a handful or two of these diamonds to show for our efforts. But the feeling of accomplishment was there because we had managed to extract these things of beauty from a place where you would expect to find nothing but solid, hard rock.

At this point, you are probably wondering what any of this has to do with the LightWave Mailing List, and perhaps you are even thinking that the effects of alcohol and sunstroke are still lingering. Well, it is quite simple. Reading the messages that come from the LightWave mailing list every day is sometimes a LOT like banging against hard rock with a sledge-hammer. In an average day, you will receive more than 100 messages, and many of them will leave your head spinning. But you can be certain that you are going to find at least a couple of messages that really help you and occasionally, maybe 4 or 5 times a week, you will come across a true gem of a message that REALLY teaches you something new and inspires you to try new things in LightWave.

I will tell you right now that the LightWave Mailing List is not for everyone. Many people get discouraged by the amount of "noise" that they have to deal with when subscribed, and many other people find that even though the list helps them a lot, they just cannot keep up with the large number of messages they receive every day. It is not uncommon for people's e-mail accounts to fill up and shut down until they empty the waiting mail. (This is especially true for people with AOL e-mail accounts!)

I have been running the LightWave Mailing List (and other lists) for almost 3 years now, and in that time it has really grown in popularity, much like LightWave in general. Right now there are approximately 1,400 people subscribed to the list from around the world, and that number is growing steadily. I receive subscription requests from maybe 40 or 50 people each week, and at least 90% of those new subscribers leave the list after only a few days. Like I said, it is not for everyone.

To help people benefit from the LightWave Mailing List without getting buried in e-mail, a Digest option can be turned on, so that all of the day's messages will be archived into one or two BIG pieces of e-mail that will be delivered each day. Some people find this useful, but I feel that a much better option is just use an e-mail program that will allow you to filter the LightWave Mailing List messages into a mail folder that is separate from the rest of your e-mail. This will allow you to read and reply to messages much more easily than when they are being sent to you in the Digest mode. There are also a couple of ways to read the LightWave Mailing List messages without receiving any e-mail whatsoever. If you have reasonably fast access to the World Wide Web, you may find the on-line mailing list archives more to your liking. This system has the added benefit of a search engine that that will help you find specific mailing list discussions. I will soon be releasing a CD-ROM (shameless plug time) that contains "the best of" the LightWave Mailing List messages from the last couple of years archived in various formats to give you quick access to tens of thousands of interesting LightWave discussions!

And finally, for those of you who do not want to break a sweat working in the mining pit, a LightWave Mailing List Frequently Asked Questions (FAQ) page is also available. This page presents a large number of questions and answers that are discussed regularly on the list, and you are practically guaranteed to find a LOT of useful info there. However, you do have to remember that this page represents only a small fraction of the interesting and entertaining topics discussed on the LightWave Mailing List.

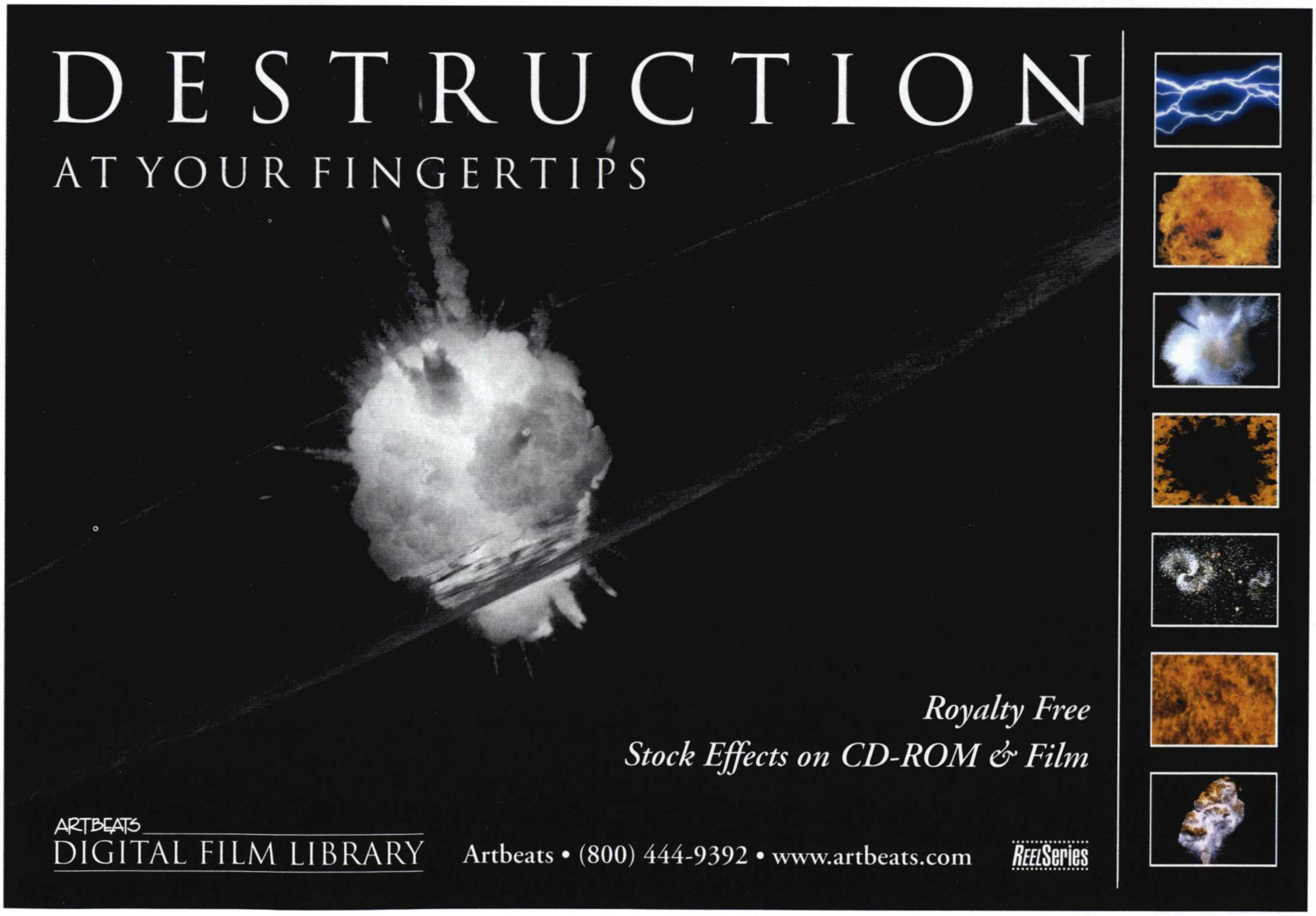
If you do decide to tap into the LightWave Mailing List, you will not only be rewarded with informative and inspiring LightWave tips, but you will become a member of a (usually) friendly and helpful group. There have been many cases where desperate LightWave users were in need of a quick answer to something they were having trouble with, and usually within a few hours after they asked for help, there were many useful replies waiting for them! This sort of quick interaction between LightWave users from all over the world makes the mailing list an invaluable resource, and in many ways it is like a close-knit community or family. We help each other; we fight with each other; we even have a monthly contest where we compete against each other!

For more information about the LightWave Mailing List and any of the various resources connected to the mailing list, send e-mail directly to me at dwarner@tv3d.com, or visit my web site by pointing your web browser to http://www.tv3d.com

I look forward to seeing you there...and remember to bring your own beer! ;-)

David Warner dwarner@albany.net http://www.albany.net/~dwarner <--- On-line Mailing List Archives

For more discussion of this topic, circle Reader Service #115

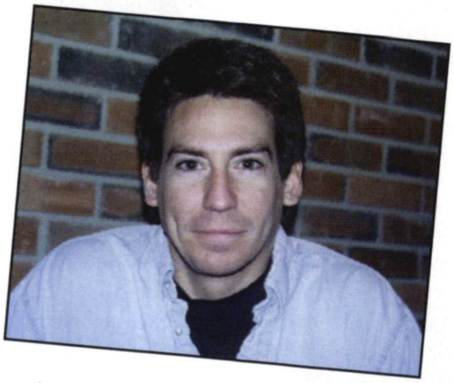


Reader Service #105

• A VISIT WITH THE ROCKETMAN HIMSELF

PROFILER

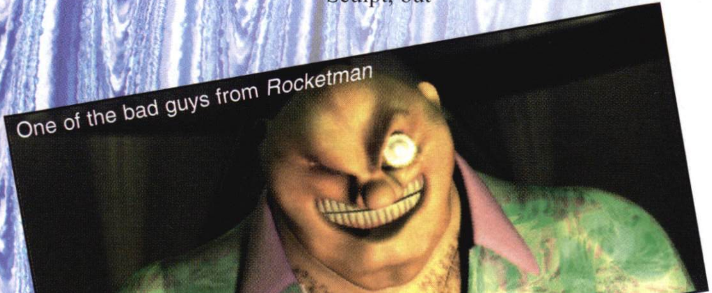
by Tiffani Banaszak



Welcome to the debut of Profiler, where you will get to meet some of the industry's most important characters. We will be visiting people from top Hollywood studios all the way to home-based animators, from names we are all familiar with to new, upcoming faces just breaking into LightWave animation. We will even get to visit some schools that focus on LightWave training, as well as a dealer or two that provides top support to their clientele. This column is about the people in the LightWave community; those who make it what it is: full of quality, a large and growing populous, and like a family.

In this issue, I decided to find an animator that everyone knows of, but isn't part of the "Hollywood" scene. After weeks of making lists of names, people whom I had met and those I had only read about, I decided on Mark Thompson of The Big Machine. Mark is perhaps more recently known for *Rocketman*, a theatrical film trailer that NewTek aired at Siggraph last summer to promote the Sun version of LightWave. He also had a following many years ago for his Fred Floaty animations.

Mark, a Chicago native, graduated in 1982 from the University of Illinois. He majored in Electrical Engineering, and wound up designing hardware for RCA in Pennsylvania. A far cry from animation, you might say. But never fear; in 1985 Mark purchased an Amiga 1000. His first foray into animation was with Sculpt, but after moving up to the Amiga 2000 with a Toaster, Mark discovered LightWave. He has never looked back.



One of the bad guys from *Rocketman*

In 1995, Mark joined the PC brotherhood when he formed Fusion Films with Paul Griswold. Their first machine was a jury-rigged 275mhz DEC Alpha, which Mark built and linked to his Amiga (those were the days). In 1997, Mark and Paul parted company, and out of the ashes The Big Machine was born. Jeff Goldman, cofounder of The Big Machine and the director of *Rocketman*, shares Mark's creative passion.

The Big Machine, with Mark at its head, has grown in scope, recently moving into a 1000 square foot facility in Nashua, New Hampshire, right outside of Boston. They employ several full-time animators, as well as many freelance artists. Having just finished a series of 8 commercials for The X-Men Wolverine toys, they are now beginning a job for NASA doing simulations. They are bidding on a job that requires "photorealistic dinosaurs" (secret stuff). Mark is also completing work on some series concepts for comic book companies looking to market their cartoons.

Mark is also marketing a 'toon of his own, *Rocketman*. He hopes to see *Rocketman* become a regular series aimed at the adult sci-fi genre. *Rocketman* was created for several reasons. First, to market LightWave as a viable Sun animation tool. And secondly, it was a test for LightWave and other Sun software to discover the quality and comparability with other platforms.

Rocketman was perhaps the longest endeavor for Mark, taking over 1 man-year of labor between himself and Jeff Goldman. All told there were 7 animators working on the project at one time or another. Those of us that had the pleasure of viewing *Rocketman* at Siggraph last year were quite impressed with the quality and realism of the animation, as well as the humorous delivery.

When Mark is not animating or running his company, he likes to get away to Jackson Hole and Grand Targee for some winter fun. He is an avid skier, and regularly travels the country in search of hills to conquer. When

Wolverine from the recently completed X-Men commercials .

the sun finally comes out for the summer, Mark and Denise, his wife of 8 years, enjoy visiting beaches and soaking up rays. They recently took a week-long cruise on a 47' sailboat around the Barry Islands in the Bahamas. They caught all of their meals, and spent many hours diving for conch and observing the beautiful coral reefs.

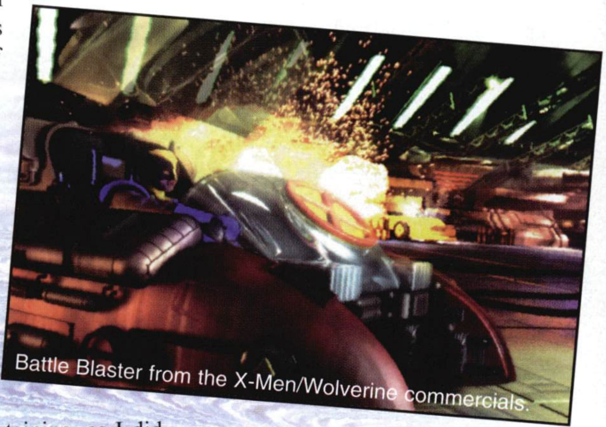
As I asked Mark my last question, "If you could change one thing in your life, what would it be?", he took a long pause, and then a longer one. "I'll have to think about it. Can I email you?" Mark, you are truly a thoughtful human being. No quick, trite answers like "world peace" or "stop pollution" for Mark. The answer took 6 days and some soul searching.

Battle scene from *Rocketman*

His answer was lengthy. “Back in '92 when I created my second major animation "Fred Floaty', I was presented with various opportunities to move to the west coast to do animation. But because of my dislike of Los Angeles and the very poor pay animators commanded back then, I chose to stick it out in New England and incrementally build my own company. In hindsight, that was the beginning of a very pivotal period for animation in Hollywood, and it would have been very advantageous to be there for it. Secondly, the number of contacts I would have made had I gone to LA would have been tremendous. So if I could change anything, I would have hit the west coast back in '92 for a few years and then returned to New England to start up my company. Gaining recognition amongst the high-end clientele has always been our most difficult hurdle, and a few years in Hollywood could have really helped out in that area.”

Train from *Rocketman*

Mark is a superb animator with a vivid imagination and a strong sense of humor. You will probably see him hanging around NewTek's booth at some of the larger shows this year. If you bump into him, ask him about the dog he should have gotten, the name of his *first* award-winning film, and let him know how much Hollywood misses him. I am sure you will find him entertaining, as I did. Mark, thank you for an enjoyable interview, and I hope you get that dog someday!



Battle Blaster from the X-Men/Wolverine commercials.

LightWave Multi-platform

by Jason Goodman

Since its inception in or around 1989, LightWave 3D has pushed the price/performance envelope. Perhaps one of LightWave's greatest strengths is its high degree of cross platform compatibility.

One of the more recent additions to the long and growing list of LightWave savvy OS' is the Mac OS. A long standing favorite of graphics professionals, the Macintosh sports a friendly user environment and a fanatical following rivaled only by the beloved Amiga. This avid following is evidenced by the remarkable sales of LightWave for PowerMac, and NewTek's commitment to these users.

Before LightWave, Mac users had many choices for creating 3D content. They could have used inadequate, unstable, reasonably priced, user friendly tools, or they could have taken out a loan and gotten ElectricImage. Now that NewTek has thrown its 3D glove into the arena, users have a more suitable choice for professional, broadcast and motion picture animation.

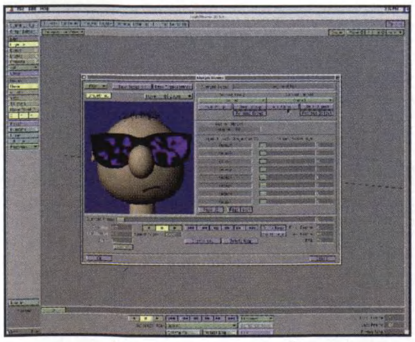
While many early adopters felt the sting of riding the bleeding edge of software technology, NewTek has worked consistently to deliver effective updates that have eased the problems that plagued earlier releases. Do not misunderstand; Mac LightWave may not be completely bulletproof, but Photoshop, Premiere, Quark and OS' prior to 8 do their fair share of crashing as well. And to all of you skeptical Windows 95/NT users, I am a multiplatform guy living in a multiplatform world, and no application on any OS is safe from digital narcolepsy. I find applications, including LightWave, crash under Windows as often as any others.

As I write this, the latest version of Mac LW 5.5 is build 247 for Modeler, and version 5.5X for Layout. You can use the Get Info command in the File menu to check your versions. If you are not up to these revs, call NewTek and pester them to update their new web site.

LightWave 5.5 for PowerMac exhibits a high degree of stability, as well as exceptional compatibility with PC and other versions of the program. The one complaint from animators in our studio was that Modeler performed much more slowly than the PC version, even on Pentium processors running at half the clock speeds of comparable PowerMacs. An informal interview with NewTek's Mac programmer Matt Craig revealed that many of these issues are related to Apple's implementation of Quick Draw 3D (the Mac alternative to Open GL) and ways in which the Mac handles screen drawing routines. Matt assured me that he is hard at work on these issues, and he has already made significant headway, improving Modeler screen redraw by over 300%.

I highly recommend the use of a 128 bit graphics accelerator from manufacturers like IXMicro or Radius. These cards seem to be even more effective than some "Quick Draw 3D accelerators." Incidentally, our test Macs (the ones our animators cited as being much slower for Modeling) were not equipped with any graphics acceleration whatsoever, while comparison PCs had Dynamic Pictures' Oxygen cards installed.

Layout on the Mac is a sheer pleasure to work with. NewTek's Don Ballance reports that speedy G3 Macs out-perform 500mhz Alphas when rendering Layout scenes. The latest version even sports many features not yet found in the PC or other versions. These include more convenient access to plug-ins, (including some that do not appear on the PC at all), more advanced stereoscopic rendering, support for additional file formats including Quicktime and 3DMF, and a greatly improved interface to Particle Storm Lite. Professional animator Jeffery Odell told me that the Mac version just feels better. Plug-ins and windows pop up more readily, and the overall look is a lot nicer. Jeff also liked the ability to utilize the MacOS' built in multiple monitor support. This allows Mac animators to spread themselves out a bit.



A screen capture of LightWave for Macintosh showing Morph Gizmo in use.

Mac LightWave has undergone some growing pains, but I say it is through puberty and ready to be taken seriously by the Macintosh 3D community. It certainly offers Mac users a highly sophisticated alternative to the comparatively limited selection of 3D applications currently available on this platform.

On another note, an exciting new development from NewTek is the release of Inspire 3D for Mac OS. Inspire is a close cousin to LightWave that will appeal to cost conscious users. If you love LightWave and you love your Mac, you should find this to be a marriage made in Topeka by way of Texas. And I think there exists somewhere a saying about Texas being close to heaven. (You can do the math on that one).

Next issue we will discuss ways to improve the overall stability of your Mac and improve LightWave's performance by optimizing your OS through Extensions management. Until then, keep Wavin', save often, and don't have any sharp objects nearby when your system hangs.

AUTHOR PROFILE

Jason Goodman is CEO of the New York based multimedia production company MSD, creators of the LENY CD ROM series, and LightWave Expo New York. Contact Jason via email at Jason@msd-nyc.com.

THE POWER to Explore your IMAGINATION

Customer satisfaction is our strength at Flight Technologies. We understand that our customers rely on us. We take extreme measures to understand your needs, goals and desires. We answer your questions, make solid recommendations, and of course, we build great systems!

"M&M's"® Brand Characters © Mars, Incorporated 1995 Image courtesy of Will Vinton Studios. Frames created and rendered with LightWave 3D™ on Flight Workstations.

Flight Workstations come with a 30-day money back guarantee. Call us to see what system best meets your needs!

Pentium II Workstation!

Single or Dual 300MHz Pentium II

Utilizing one or two Pentium II™ processors, the KittyHawk II offers outstanding power and flexibility. Optimized for Windows NT™ applications, built with the fastest components available today *and* aggressively priced, why settle for anything less?

KittyHawk II Basic Configuration:
- Dual Pentium II System
- Single 300MHz Pentium II installed
- Solid Steel Chassis, 400 Watt PS
- 64 mb. RAM
- On-board SCSI-III
- Ultra/WIDE SCSI w/ 4 gig. WIDE Hard Drive
- 24X CD-ROM
- 3.5 Floppy, Keyboard, PS/2 Mouse
- 10/100 Ethernet
- AGP Diamond FireGL Pro w/ 8mb SGRAM
- SoundBlaster 64 PNP
- Yamaha System 35 Subwoofer/Speakers
- ViewSonic/Optiquest 17" Monitor
- Windows NT Workstation

$4,495 or $160/mo.*

Dual Pentium II Power! NEW

Single or Dual 333MHz Pentium II

Powered by dual 333MHz Pentium II™ processors, this KittyHawk II system offers everything a professional animator needs. This KittyHawk II package provides the ultimate combination of components to provide explosive performance for the ultimate animation workstation!

KittyHawk II Professional Configuration:
- Dual Pentium II System
- Two 333MHz Pentium II processors
- Solid Steel Chassis
- 400 Watt PS
- 256 mb. RAM
- Ultra/WIDE SCSI
- Two 4 gig. WIDE Hard Drives
- 24X CD-ROM
- Floppy, Keyboard, Mouse
- 10/100 Ethernet
- Oxygen 402 OpenGL Accelerator Card
- Sounblaster Sound Card
- Yamaha System 35 Subwoofer/Speakers
- ViewSonic/Optiquest 21" Monitor
- Windows NT Workstation

$10,099 or $355/mo.*

Call Us Today!

(800) 597-4470
www.flightinc.com
sales@flightinc.com

ALPHA GENERATION *We sell and support Alpha-based workstations as well. If you are using Lightwave or SoftImage|3D, call us for the fastest Windows NT Workstation on the planet!*

intel inside PENTIUM® PRO PROCESSOR

FLIGHT Technologies, Inc.

MMX™ intel inside pentium® II PROCESSOR

Contact us at: Flight Technologies, Inc. • 9164 West 57th Street Merriam, Kansas 66203 • (913) 236-7550 • (913) 236-7429 fax • http://www.flightinc.com/ • sales@flightinc.com

**Prices subject to change and many options are available to customize these systems to your needs. Monthly rate based on 36/mo. fair-market value lease. Call us for latest rates and pricing for your configuration. Flight Technologies, Inc., the Flight logo, KittyHawk Pro and KittyHawk II are trademarks of Flight Technologies, Inc. All other products are trademarks or registered trademarks of their respective companies.*

Reader Service #103

PROGRAMMER'S HIDEWAY

by Brenden Mecleary

Given that LightWave is one of the most popular 3D rendering programs out there, I have always been surprised that there are not more plug-ins available for it. I am not exactly sure why that is. Maybe there are fewer programmers using LightWave than 3DS MAX. Maybe the LightWave SDK documentation is a bit tough to read. Maybe LightWave is so full-featured right out of the box that it does not need as much improvement as the other programs!

Whatever the reason, I am going to see if I can't get more of you "closet programmers" out there (don't deny it - you know who you are!) into your development environments and writing those plug-ins. It has only been about three months since I wrote my first LightWave plug-in in C++, and about six months since I wrote my first program in C++. So take it from the ultimate newbie - writing LightWave plug-ins is not hard. To show just how easy it is, I will take you through the creation of a LightWave plug-in in C++.

Before I start, be forewarned that I expect you to have a reasonable grasp of C++. This is not a language tutorial. Anyone who emails me and asks what a pointer is will be offered a seat in the comfy chair. This tutorial covers plug-in writing for Intel machines. I assume, though, that the process is pretty similar on other machines.

There are so many aspects of LightWave plug-ins that I cannot possibly cover all of them in a single article. Since Layout plug-ins are the most complex (and fun!), we will tackle one first. Specifically, we will make an image filter that will put a color gradient overlay on the screen. Not particularly useful, perhaps, but we can use it to cover the main points of the LightWave plug-in architecture, and how to add an interface to your plug-in. Plug-ins are DLL files (renamed to have a ".p" extension) that LightWave will load when they are activated. First, download and install the LightWave SDK (and LWPanels SDK) from NewTek (www.newtek.com), and then we can jump right into the source code. I will explain along the way.

```
//Includes
#include <splug.h>
#include <moni.h>
#include <lwran.h>
#include <lwpanel.h>
#include <strstrea.h>

//Typedefs
typedef unsigned short int USHORT;

//Defines
#define NULL 0


//Global constant (reduces string table size)
const char * const SPACE = " ";
```

Here are the files you will need to include, and a typedef / define for convenience. In addition to the files included above, you will need to add serv.def to your project, and link with serv_w.obj and server.lib. In fact, do not link with anything else; it will just take up space. The first four included files are required for LightWave Layout plug-ins. The last include is part of the standard C++ library, which we will use for the Save and Load functions. I added the global SPACE constant so that we can add spaces to strings without wasting storage in the string table.

Now, being good C++ programmers, let's put in the prototypes for our plug-in's functions:

```
//Function prototypes
extern "C"
{
int GRADActivate(long, GlobalFunc *, void *, void *);
int GRADInterface(long, GlobalFunc *, LWInstance, void *);
}

LWInstance GRADcreate(LWError *);
Void GRADdestroy(LWInstance);
LWError GRADcopy(LWInstance, LWInstance);
LWError GRADload(LWInstance, const LWLoadState *);
LWError GRADsave(LWInstance, const LWSaveState *);
const char* GRADdescln(LWInstance);
const LWItemID* GRADuseItems(LWInstance);
void GRADchangeID(LWInstance, const LWItemID *);
void GRADprocess(LWInstance, const FilterAccess *);
unsigned int GRADflags(LWInstance);
```

We declared two functions to be C-linked to prevent name decoration. Because the libraries that NewTek provides were created using C-linkage (and the general nature of DLL files), there are difficuties if this is not done. Declaring the Activation functions (and one other global, which I will cover later) to be C-linked lets us have all the benefits of C++ coding when writing our plug-ins. (So you know, the Interface function is, for all practical purposes, another activation function.) Because no name decoration means no function overloading, you should get in the practice of prefixing your plug-in's functions with something to tell what plug-in they refer to. Otherwise having multiple plug-ins in the same ".p" file will not be possible. (After all, which "Activate" would LightWave use?)

Next, we will declare a small class to hold the data our plug-in uses. When you first activate the plug-in, LightWave will call the GRADActivate function. It will then call GRADcreate, which must allocate memory for this plug-in on the heap and then give a pointer to it back to LightWave. But, I am getting ahead of myself. Here is the class that we will use to store data:

```
//Instance class
class GRADIENT
{
public:
        //Gradient colors at the top and bottom of the image
        USHORT cTop[3], cBottom[3];

private:
};
```

Notice that all the data members are declared to be public. C++ super-purists will not like this, but there is no data here that needs to be hidden from anyone, so we will forego the trouble of calling functions each time we need a variable. Aside from that, the class Declaration is pretty straightforward.

Next, let's insert the ServerRecord structure. Basically, this is a big array that tells LightWave about all the plug-ins in the file. There is another method, but this way you can declare more than one plug-in (or each plug-in and its respective interface), all at once:

```
//The ServerRecord structure identifies all the plug-ins in this file
to //Lightwave.
extern "C"
{
ServerRecord ServerDesc[] =
{
{ "ImageFilterHandler", "LightWavin_gradient", GRADActivate },
{ "ImageFilterInterface", "LightWavin_gradient", GRADInterface
},
{ NULL }
};
}
```

This must also be C-linked. Each line contains three pieces of information, and the last entry in the array is NULL.

The first constant in the line is the type of plug-in. Each type of Layout plug-in (except Generic plug-ins and some of the more obscure types) has a "Handler" and "Interface" type. If you leave out the "Interface" type, then LightWave will show the Options button next to your plug-in ghosted out. The next item in the line is the name that goes into the lw.cfg file and shows up in the drop-down list of plug-ins.

Do not use any spaces here, or LightWave will not be able to find your plug-in later on. Finally, the last part of the line is the name of the "Activation" function, which is the first function that will get called when you select the plug-in from LightWave.

One last thing before we get to the actual functions: a few globals to make it easier to add the interface for our plug-in:

```
//These globals allow you to use the macros to add controls in the
interface.
static LWPanControlDesc   desc;

static LWValue ival={LWT_INTEGER},ivecval={LWT_VINT},
fval={LWT_FLOAT},fvecval={LWT_VFLOAT},sval={LWT_STR
ING};
```

Okay. The globals are out of the way, so now we can write the activation function:

```
//Activation function. This function tells Lightwave where to find
all the
//functions it needs to run the plug-in.
int GRADActivate(
        long                version,
        GlobalFunc          *global,
        void                *locPtr,
        void                *data
        )
{
ImageFilterHandler *local = reinterpret_cast<ImageFilterHandler
*>(locPtr);

        if(version != 2) return AFUNC_BADVERSION;
        local->create = GRADcreate;
        local->destroy = GRADdestroy;
        local->copy= GRADcopy;
        local->load= GRADload;
        local->save= GRADsave;
        local->descln= GRADdescln;

        local->process= GRADprocess;
        local->flags= GRADflags;

        return AFUNC_OK;
}
```

The first thing we do when the Activate function gets called is to cast the void pointer "locPtr" to the proper type of structure. Each plug-in type has its own structure and specific functions. For example, an image filter uses a "process" function, while surface shaders use "evaluate" functions. Note that if we were using C to write this plug-in, we could simply change the parameter list to represent the right structure, and not have to cast it. C++ will not let us do that, though, but it is a small enough price to pay for the other conveniences.

The next thing we do is version checking. Plug-ins for LightWave 5.0 should be version 2. The 5.5 SDK bumped that up to 3. Each new version has a few extra members in the structure, so be sure to only use the ones in the version you have. When LightWave calls the Activation function, it will first try the current version number. If that does not work (and you return AFUNC_BADVERSION), it will go down one version and try again. LightWave will do this all the way down to version 1, if necessary.

Having checked the version number, the only thing left to do is put pointers to all of our functions into the structure and return AFUNC_OK, which tells LightWave that the function succeeded.

The next function we will define is the Creation function. This is a nice, short little function. It allocates memory, then returns a pointer to that memory. Even if your plug-in uses no instance data, you must allocate memory and return a valid pointer to LightWave.

Returning NULL from the Creation function will generate an error.

```
//Creation function
LWInstance GRADcreate(LWError *error)
{
        GRADIENT *inst = new GRADIENT;
        return(reinterpret_cast<LWInstance>(inst));
}
```

By the way, get used to seeing the reinterpret_cast Declaration quite often. A regular cast will do just as well, but I am already in the habit of using the new casts, so that is what I will use here. Also, FYI, a LWInstance is simply a void pointer.

Next, we will create the destroy function. This is also quite simple; just delete the pointer we've got:

```
//Destroy function
void GRADdestroy(LWInstance inst)
{
        if(inst)
        {
GRADIENT *instToDelete = reinterpret_cast<GRADIENT
*>(inst);
        delete instToDelete;
        return;
        }
        return;
}
```

The next function on the list (this is going pretty fast, isn't it?) is the Copy function. This is another simple function. Keep in mind that because we are just using a few standard types, we did not bother to write an assignment operator for our instance class. However, if there were user defined types in there that had allocated memory (strings, for example), we would probably need to write one.

```
//Copy function
LWError GRADcopy(LWInstance from, LWInstance to)
{
        GRADIENT *first, *second;
        first = reinterpret_cast<GRADIENT *>(from);
        second = reinterpret_cast<GRADIENT *>(to);
        *second = *first;
        return NULL;
}
```

Next up is the Save function. When writing plug-ins whose data is not stored in the scene file, (i.e. shader plug-ins), stick to ASCII text data for storing plug-in information. The primary reason for this is portability. Unless you enjoy tinkering with byte orders, using ASCII text will make sure that nobody will have problems reading your plug-in data.

```
//Save function.
LWError GRADsave(LWInstance inst, const LWSaveState *lwss)
{
        //Convert the instance pointer
        GRADIENT *defInst = reinterpret_cast<GRADI-
ENT*>(inst);

        //Allocate the output string
        char cOutputString[32];

        //Enter the data
        ostrstream ostr(cOutputString, 32);

        ostr << defInst->cTop[0] << SPACE << defInst->cTop[1]
<< SPACE;
ostr << defInst->cTop[2];
        ostr << SPACE << defInst->cBottom[0] << SPACE <<
defInst->cBottom[1];
        ostr << SPACE << defInst->cBottom[2] << ends;

        //Write the data to the scene file
        lwss->write(lwss->writeData, cOutputString, 32);

        return NULL;
}
```

First we convert the pointer LightWave gives us (inst) into one we can use. Then we allocate a character string with enough space to hold all the information. We fill that in with all the color data, then write it out and return OK. If you have not used them yet, the string streams are the C++ equivalent of sprintf and sscanf, but they are much easier to use, and type-safe.

Of course, now that we can write data, we need a way to read it, so here is the Load function:

```
//Load function
LWError GRADload(LWInstance inst, const LWLoadState *lwls)
{
        //Convert the instance pointer
        GRADIENT *defInst = reinterpret_cast<GRADI-
ENT*>(inst);

        //Allocate the input string
        char cInputString[32];

        //Get the data from the scene file
        lwls->read(lwls->readData, cInputString, 32);

        //Extract the data into the instance
        istrstream istr(cInputString, 32);
        istr >> defInst->cTop[0] >> defInst->cTop[1] >> defInst-
>cTop[2];
        istr >> defInst->cBottom[0] >> defInst->cBottom[1];
istr >> defInst->cBottom[2];

        return NULL;
}
```

Relatively simple, right? Basically, the Load function does the same thing as the Save function, but in reverse. We convert the pointer, read the data from the scene file, and then convert it from ASCII format into usable numbers. In the LightWave Load and Save functions, the first parameter is a pointer to another function, the second parameter is the buffer they go into, and the last parameter is the maximum number of characters to read or write.

The next function, descln, simply gives a more reasonable name to the plug-in once it has been activated:

```
//Returns a descriptive line for the plug-in
const char* GRADdescln(LWInstance inst)
{
        return "Lightwavin' Plug-in Tutorial";
}
```

Because this particular function returns a string constant (that is, it is stored in the string table in the plug-in file), you can do this. However, LightWave needs whatever text you give it to stay in memory. So, if you want to have the line contain information about your plug-in (say, the current settings), you will need to declare it in such a way that it will not go out of scope when the function returns. A static string or character array will do the trick.

The Flags function tells LightWave which special buffers the plug-in will need. You have access to quite a few: Shading, Shadow, Depth (Z-buffer), Raw Surface Colors, etc. In this case, we will not be needing any of them. LightWave always gives you the RGBA buffers of the image, without requiring you to ask. In fact, you cannot ask, so do not try.

Something important to remember is how you return Flags. Usually, they are numbers that you just bitwise OR for the return value. For example, a surface shader may have a line like this:

```
return (LWSHF_COLOR | LWSHF_TRANSP);
```

However, an image filter's flags are offsets. That is, the flags look like this:

```
return ( (1<<LWBUF_DEPTH) | (1<<LWBUF_SHADING) )
```

Finally, keep in mind that it takes a lot of memory for the LightWave buffers, so only ask for those that you need. Since we will not need anything but the default buffers, we will just return NULL:

```
//Tells LW which buffers the plug-in will need unsigned int

GRADflags(LWInstance inst)
{
	return NULL;
}
```

Okay! I have saved the best for last - we have got only two functions left, and they are the most fun! We need to write the Process function, which will do the actual image processing, and the Interface function, which...well...makes the interface. Get yourself a cup of coffee (go ahead, you have earned it!), and then let's write the cool parts.

Back? Let's write the last part of the "core" plug-in, the Process function:

```
//Process - The fun stuff!
void	GRADprocess(LWInstance inst, const FilterAccess *fa)
{
	//Convert the instance
	GRADIENT *defInst = reinterpret_cast<GRADI-
ENT*>(inst);

	//BufferValue pointers to process the image
	BufferValue out[3], *r, *g, *b, *a;

	//Create the monitor for our use
	Monitor *mon = fa->monitor;

	//Initialize the monitor
	MON_INIT (mon, fa->height);

	//Loop through the lines in the image
	for (USHORT i = 0; i < fa->height; i++) {
		r = fa->bufLine(LWBUF_RED,   i);
		g = fa->bufLine(LWBUF_GREEN, i);
		b = fa->bufLine(LWBUF_BLUE,  i);
		a = fa->bufLine(LWBUF_ALPHA, i);

	//This is the current percentage of the image that this line
is
		double dPercent = i;
		dPercent /= fa->height;

#pragma warning(disable : 4244)

	//Blend the gradient values
	unsigned char currR = (defInst->cTop[0] * (1 -
dPercent)) +
		(defInst->cBottom[0] * dPercent );
	unsigned char currG = (defInst->cTop[1] * (1 -
dPercent)) +
		(defInst->cBottom[1] * dPercent );
	unsigned char currB = (defInst->cTop[2] * (1 -
dPercent)) +
		(defInst->cBottom[2] * dPercent );

#pragma warning(default : 4244)

	//Divide the gradient by two (for a 50% blend with the
image)
		currR /= 2;
		currG /= 2;
		currB /= 2;

	//Loop through the pixels in the line
		for (USHORT j = 0; j < fa->width; j++)
		{
			out[0] = (r[j] / 2) + currR;
			out[1] = (g[j] / 2) + currG;
			out[2] = (b[j] / 2) + currB;

			//Set the final pixels
			fa->setRGB(j, i, out);
			fa->setAlpha(j, i, a[i]);
		}
		//Step the monitor every 10th line
		if ((i % 10) == 0) MON_INCR (mon, 10);
	}

	//Kill the monitor
	MON_DONE (mon);

	return;
}
```

To start, we cast the instance pointer into the pointer we need for our class. Next, we allocate some variables of type BufferValue, which is a typedef for unsigned char. Conveniently, this number is what we need for a true-color image: between 0 and 255. Be careful when doing math directly on BufferValue types. There is a good chance you will go over 255 or below 0 and get some funky colors in your image. It is better to cast it to an integer first, then bounds-check later on. The first BufferValue variables we allocate are an array. They will hold our output values. The last four are pointers to a line of pixels.

Next, we create a monitor. We get it from within the FilterAccess structure. Then, we initialize it with the number of lines in the image. After that, we enter the loops. The first loop goes through all the lines in the image. On each line, we call bufLine to get a pointer to the line of pixels. There is also a fltLine function, which you use when you are getting the buffers for depth.

Before we enter the block that loops through each pixel in the line, we do a little bit of math on the gradient values. First, we get the percentage of the image we are at. Then, we multiply that with the gradient colors. Because the percentage should always be between 1 and 0, there is no danger of overflowing the data type. Next, we add the gradient colors together to get the result. Then, because we are blending the gradient with the image half-and-half, we divide the colors by two.

By the way, if you are wondering about those pragmas, they prevent the compiler from griping about my mathematics. (Specifically, I am assigning a double result to an unsigned character).

At last, we arrive at the block that changes the pixels and gives them back to LightWave. It is actually somewhat small. First, we divide the image colors by two (50% blend, remember?), and add the gradient color. Then, we call setRGB and setAlpha to give the pixels back to LightWave. The parameters are the X value, the Y value, and the output values. In the case of setRGB, it expects an array. SetAlpha expects a single BufferValue. It is important to note at this point that you must set all the pixels in the image, whether you make any changes or not. If you do not, you will get a nice, big, blank image. It may look pretty if you are doing a scene in a closed room with no lights or windows, but is pretty useless otherwise. So, make sure to set all the pixels, even if you run into an error.

Back outside the loop, we have an "if statement" that, once every ten lines (the only time that the line number modulo 10 equals 0), will update the monitor. This is significant in that it gives the user a chance to see the progress, and abort if necessary. Calling it every line is overkill, so 10 is a nice round number.

Finally, outside both loops, we call MON_DONE to stop the monitor. We return, and that's it for the process function!

Now, on to the Interface function! As I mentioned above, the Interface function is, for all practical purposes, another Activate function. You can have a plug-in without an Interface function, but not without an Activate function. Also, do not feel limited to using LWPanels.p if you do not want to. You are free to create the interface any way you want.

Windows would be more than happy to let you create a window for your interface. However, in the interest of consistency, we will use the LightWave-style interface. Here is the Interface function for our plug-in:

```
//The Interface function
int GRADInterface (
        long            version,
        GlobalFunc      *global,
        LWInstance      inst,
        void            *serverData
        )
{
        //Variables used to work the panels
        LWPanelFuncs    *panl;
        LWPanelID                       panID;

        LWControl                       *ctrl;
        MessageFuncs    *message;

        GRADIENT *defInst = reinterpret_cast<GRADIENT
*>(inst);

//Get the message function ready.
message = reinterpret_cast<MessageFuncs*>(global("Info
Messages", GFUSE_TRANSIENT));
        if (!message )
                return AFUNC_BADGLOBAL;

//Get the panel function ready.
panl=reinterpret_cast<LWPanelFuncs*>(global(PANEL_SER-
VICES_NAME, GFUSE_TRANSIENT));

        if(!panl)
        {
                (*message->error)("Unable to activate global
"PANEL_SERVICES_NAME, "     please add plug-in lwpanels.p"
);
                return AFUNC_BADGLOBAL;
        }

//This creates the panel in memory.
panID = panl->create("Lightwavin' Plug-in Tutorial Interface",
NULL);

//Add and set controls now.
{
        RGB_CTL(panl, panID, "Color at top of image");
        ctrl = panl->nextControl(panID, NULL);
SET_IVEC(ctrl, defInst->cTop[0], defInst->cTop[1], defInst-
>cTop[2]);
        }
        {
        RGB_CTL(panl, panID, "Color at bottom of image");
        ctrl = panl->nextControl(panID, ctrl);
        SET_IVEC(ctrl, defInst->cBottom[0], defInst-
>cBottom[1], defInst->cBottom[2]);
        }

        //This displays the panel. Returns 1 for OK, 0 otherwise.
        if( panl->open(panID, PANF_BLOCKING |
PANF_CANCEL) )
        {
ctrl = panl->nextControl(panID, NULL);
GET_IVEC(ctrl, defInst->cTop[0], defInst->cTop[1], defInst-
>cTop[2]);

                ctrl = panl->nextControl(panID, ctrl);
                GET_IVEC(ctrl, defInst->cBottom[0], defInst-
>cBottom[1], defInst->cBottom[2]);
        }

//This kills the panel.
        panl->destroy(panID);

        return AFUNC_OK;
}
```

...I know that is a bit messy, but hopefully it is still legible. The first thing we do in the Interface function is declare a few variables, and convert our instance pointer. Then we call a LightWave Global function (provided as a parameter) to get the MessageFuncs pointer, and the PanelFuncs pointer. The MessageFuncs pointer is used to provide little messages (the LightWave equivalent of Windows' MessageBox function) to the user. In theory, you can get this whenever you want, even during render. (However, if you do, you will probably make the user really mad...) The PanelFuncs pointer is used to operate on the LightWave panels, and we will be using that quite a bit.

Back to the interface. We create a panel by calling panl->create with the name of the panel, and NULL as the second parameter. The second parameter is a void pointer. When working with Panels, there are a number of callback functions you can define (if you want to). Whenever one of those functions gets called, you are given that pointer. If you have some data that you would like to pass to those callbacks, this is an easy way to avoid resorting to global-scope variables. In case you are wondering, you do not have to set the panel size unless you want to, so that is why we did not. It will automatically size to fit the controls you put on there.

The next step is to add and set the controls on the panel. I know this part may look a bit strange, but separating it into blocks makes clearer (to me, at least) what is going on. Inside each block, we add and set a separate control. Note the commands that are in capital letters. Those are preprocessor macros that LWPanels.h gives you. The globals we declared above make it possible to use them. For each control, we declare its creation (first line), set the ctrl pointer to that control (second line), and set the values of that control to those in our instance.

The part where the action happens is next, where we call panl->open, with the panel ID that we want to open, and the style. PANF_BLOCKING specifies that the panel will stay on top of everything until it is closed. PANF_CANCEL specifies that the panel will have a cancel button. See how we put it inside an If statement? The open function will return true if the user clicked OK, and false if the user clicked cancel. This is a convenient way to test whether the user said OK to the changes.

Inside the If block, we set all our instance data to whatever the controls' current states are. To start, we set the ctrl pointer to the first control on the panel. Then we use another macro to get the values of that control. After that, we step the pointer (by calling nextControl with the panel ID and the current value of the pointer) and do it again. Finally, we call the Destroy function with the panel ID, and return OK. The documentation mentions that there is a close function for panels. However, do not close a panel and then destroy it. That will crash LightWave.

That is it! You are done! Build the plug-in, install it, and check out your handiwork! If your compiler is like mine, you will get a warning about the output filenames for your project. You can safely ignore that. Go ahead and play with your new creation!

So, how does it feel to be an official LightWave plug-in programmer? Now that you have some of the basics down, moving on from here should just get easier. My first plug-in (AlphaClip) was very much like this. It is not particularly complicated, but now you have some familiarity with the structure that LightWave plug-ins follow. If you did not feel like typing all that, never fear! The source code, MAKfile, and completed plug-in are all on the CD that came with this magazine.

To finish up, let me share with you some of the advice I have been given. One thing I suggest you do is make a "utilities" file. This holds all the various functions that you may need when writing LightWave plug-ins. For example, I have a function called ImgBailOut that I can use if I want to abort from an image filter. It is a stripped down image filter that essentially just sets all the pixels in the image to whatever they were before. That way, if I have a problem, I can just call that function and return. Some of the other functions I have are for blending pixels, getting the brightness of a pixel, finding objects in the scene, etc.

About the Global function pointer - there does not seem to be any way to get LightWave to give it to you whenever you need it. It only does so when it calls the Activate or Interface functions. The solution? Declare a global-scope pointer, then assign it from the Activate function. This may seem to be questionable practice, but I have done it many times and have yet to run into problems from it We did not need the Global function pointer in this plug-in, but if you ever do, now you know how it is done.

The last thing I suggest is to read (groan) the LightWave SDK documentation. I know it is oddly structured, lengthy, and a bit confusing, but give it a crack anyhow. I suggest that whenever you start a plug-in of a type you have not made before, look-up how it is done. Layout plug-ins are all covered very well in section 8 of the documentation, and the Global functions are in section 9. In those sections you will find the most useful information, from structures that LightWave provides, to functions that are specific to each type of plug-in.

And that wraps it up! Let me know what you thought about the article! I hope you learned something...did I motivate you to start writing your own plug-ins? Then what are you waiting for?

AUTHOR PROFILE

Brenden Mecleary, 17 year-old programmer, pilot, and animator, is the author of RealiTools. If there are topics you would like to see covered in Programmer's Hideaway, please email Brenden at bman2@ix.netcom.com.

For more discussion of this topic, circle Reader Service #116

What's up for next issue's Programmer's Hideaway?

Brenden will deal with LScript...a beginner's introduction and basic programming techniques. This will be followed by gradually more advanced LScript programming tutorials.

Take One

by Tiffani Banaszak

Welcome to the first episode of Take One! In this column, we will be visiting the virtual sets of major motion pictures, television shows, commercials, and computer games. All of these environments have two things in common: they do not exist anywhere outside of a computer, and the ones we will visit are created using NewTek's LightWave 3D. We will be exploring the process of creating high-end computer graphics, the machines and software used, as well as other details of the industry.

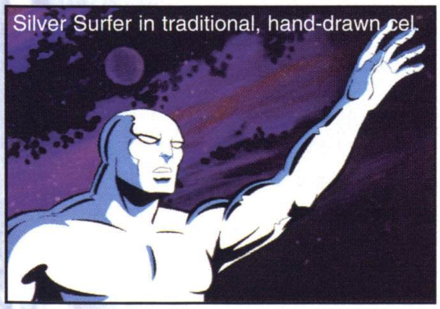
For this issue, *Keyframe Magazine* had me visit with Joe Clason of Joe's Digital Bar and Grill. The topic was *Silver Surfer*, a Saturday morning cartoon based on the Saban/Marvel comic, and this issue's cover image.

KF: Who's idea was *Silver Surfer*?

JC: *Silver Surfer* is the first of ten properties that Marvel and Saban want to produce. It was in pre-production for about a year and a half, developing the look and feel of the show.

KF: Did you read the comic before doing the series?

JC: You bet! Saban is full of comic book aficionados who know the entire lore of Marvel, and produced the storyboards we work from and that are sent to Korea for the traditional animation. Stan Lee himself has been in the studio and offered a suggestion or two about the show, since Marvel is a few buildings down from Saban. We have stacks of reference comic books, and even went to the local Comic Book Swap Meet looking for rare, old copies of *Silver Surfer*.



Silver Surfer in traditional, hand-drawn cel.

KF: What posed the largest problem in getting the results Saban wanted in the beginning?

JC: The accelerated growth with which the department grew was the biggest problem in getting the CGI on the airwaves. What began as a labor of love with a couple of borrowed machines and the most meager of budgets, turned into a ramping up of a full blown computer animation studio while they were putting together the animation for the show. I praise Dale Hendrickson, the CGI Art Director, and our Technical Direct, Don McCoy, for their incredibly long hours and stamina to get the department where it is today. LightWave 3D has always exceeded Saban's expectations, and Haim Saban himself has personally congratulated everyone on the show.

KF: How closely do you think you've gotten to capturing the original spirit of the comic?

JC: We wanted the Saturday morning show to look as if a comic book had come to life, and on those terms I feel we have come very close. The writers started with the origin of the *Silver Surfer*, and all of the characters are from the original comic books.

KF: What are the benefits of using LightWave as opposed to other 3D packages?

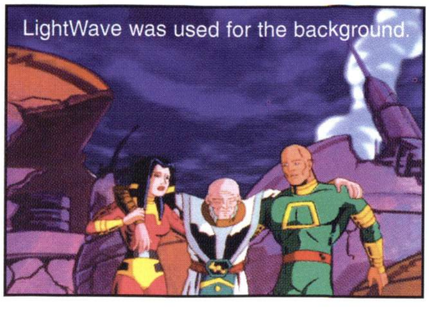
SILVER SURFER
TAKE THE RIDE!
FOX 11

Digital Joe with a Silver Surfer billboard.

JC: For *Silver Surfer*, it has to be the wonderful Cel Shader and SuperCel Shader plug-ins that Allen Hastings wrote. It's a good thing that Allen is an anime fan, or the whole project might never have happened. The Cel plug-ins produce exceptional results. MetaNURBs has also been a boon, since we animated Galactus with NURBs and Puppetmaster beginning with the third show. (You'll see a style change after episode 2 if you watch carefully!) Particle Storm was used for his eye energy and rocket exhaust, and SuperGlow put that little extra sparkle in his eye. NewTek, as well as the third party plug-in developers, have been very supportive of the project.

KF: What hardware is used for the making of *Silver Surfer*?

JC: There are twelve animating stations. Most are Carrera 533 MHz machines with 128MBs of RAM and Diamond Fire 100 OpenGL cards. There are 3 Pentium-based DPS Hollywood systems for input and output to a Digital Betacam. We use 3 DPS Perception boards for visual checks of the animation. A 72G Raid Array holds most of the data, and a dedicated PC runs WaveNet software for the rendering chores. A separate Saban machine is tied to their existing network with a T1 backbone for fast offsite connections and Internet downloads. There are two 533 MHz Alpha machines dedicat-


LightWave was used for the background.

ed to rendering. At night, all 14 Alphas contribute to the renderfarm.

KF: Other than LightWave, what software packages are relied on for creating *Silver Surfer*?

JC: LightWave 3D is used for all the 3D modeling and rendering for *Silver Surfer*. Galactus (on the cover of this issue) is the only 3D character, and he is animated with a combination of PuppetMaster and Morph Gizmo. His glow is a combination of LightWave's built-in Glow and Pontari's SuperGlow, as well as Particle Storm. Digital Fusion is used to composite the show's traditional effects and 3D effects. Some of the lightning is 3D, created with Blevin's VertiLectric, while some is created with either Digital Fusion, WinImages or MediaPaint. Most of the texture maps are created in Photoshop or painted traditionally and scanned in and then edited in Photoshop. Steamer and Particle Storm are used for the space ship exhaust, but they are stylized to look more like cartoon "Kirby" style. Of course, *Silver Surfer* is created with the Cel Shader and the SuperCel Shader

KF: Do you feel that 3D and 2D complement or fight each other? Why?

JC: Without getting on a soapbox, I feel style is everything in terms of art. Most photorealistic 3D is great, but it is just a small part of the artistic world, and frankly, not my favorite part. In photorealism, the best work possible will be totally missed by the audience. The 2D/3D debate is really moot, if you realize that the style and writing drive the story, and that is exactly what the *Silver Surfer* is all about. I would like to have done the entire show in 3D, and I think we could have, but the time and budgets wouldn't have allowed it to ever air. But *Silver Surfer* is just the tip of the iceberg in terms of sinking the way traditional animation is handled, and evolving into a cel-shaded 3D universe.

KF: How much time is spent on each episode?

JC: Each show has its individual challenges, but the schedule calls for about six animators and one compositor to produce all the animation in about two weeks. This is an average of about 150 shots per episode. The schedule means we have all the shots completed two weeks before airing. Some shots get retakes up to a couple of days before airing.

KF: How many animators are working on *Silver Surfer*?

JC: At the moment, there are eight full time animators producing *Silver Surfer*, and four support staff. Saban is doing many other projects, so the animators do other small projects as well. Some of the 3D animated bumpers are a good example of their other work.

KF: Who writes the storylines? Do you as a company have any input?

JC: Stories and storyboards are developed by Saban. The traditional art is penned in Korea, and Digital Fusion is used for compositing both the 2D with the 3D and the traditional cel with the 3D.

KF: What percentage of the work is done at your studio? What is done elsewhere? Where?

JC: We do most of the work right here at Joe's Digital Bar and Grill. We mirrored a drive so all of our scenes, objects, and other data would easily translate to the CGI department at Saban. With our ISDN line and Saban's T1, we can transfer data very quickly. Most of the compositing is done in house through Arabian or his company, Hypnotic, because the Flame sessions at the Post Group quickly became too expensive.

KF: How do you handle the difference in frame rate of traditional, hand-drawn cels, to the fps rate of LightWave?

JC: The traditional animation is done in Korea on what is known as "twos", or twelve doubled frames per second, and we render Surfer at 24 frames per second. Unfortunately, because of budget constraints, the traditional animation looks a little jerky by comparison. In Digital Fusion, the traditional and the CGI animation get composited and given a 2-3 pull up, which stretches it further to 30 fps. The traditional space pans and surfer flying are the jerkiest by comparison to the silky-smooth movements of the CGI.

The ship was modeled in LightWave.

KF: Do you think *Silver Surfer* is the most challenging job you've done?

JC: *Casper, A Spirited Beginning*, was much tougher in terms of getting the tools to do what we needed them to do. LightWave 3D and its tools are much more robust and seasoned, even since last year, making *Silver Surfer* a bit easier to do. The Cel Shader is very fast, making the render time of one to three minutes reasonable for the fast production schedule.

KF: How do you handle adding the voices?

JC: We get the voice tracks on DAT tape and Exposure sheets from the animation department of Saban. Morph Gizmo is then used to do the lip syncing.

KF: What are future plans for *Silver Surfer*?

JC: The second season's scripts are in progress, and the toys are available at all the usual places. We are off to do the next Addams Family movie for Saban after leaving Galactus in the hands of some truly great animators. Watch for some other LightWave 3D projects from Saban in the near future as well!

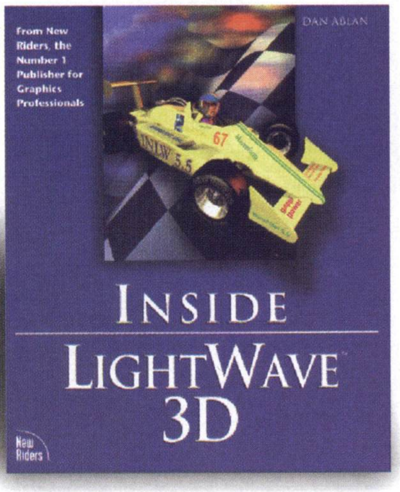
Thanks, Joe, for answering all of my questions. *Silver Surfer* can be caught on Saturday mornings at 10:30 PST.

Reviews You Can Use

- **INSIDE LIGHTWAVE 3D BOOK**
- **RELATIVITY**
- **DARK TREE TEXTURE PACKAGE**
- **EPIC MODEL LIBRARY**
- **MOTION PACK PLUG-IN**
- **FILM GRAIN PLUG-IN**

Inside LightWave 3D

One of the most notable things about LightWave users is how close-knit they are. We are in a rare community of professionals who are willing to share trade techniques. It is impossible not to notice the "team" atmosphere that has always been a part of LightWave. *Inside LightWave 3D*, published by NewRider's Publishing, is an excellent monument to all of the success and togetherness in this community. It is written by six well-known LightWave experts and edited by Dan Ablan. A well composed, comprehensive and detailed step-by-step guide, *Inside LightWave 3D* appeals to both the power-hungry advanced users and the beginner who is just getting started. This is perhaps the book's greatest strength.


From New Riders, the Number 1 Publisher for Graphics Professionals
DAN ABLAN
INSIDE
LIGHTWAVE
3D
New Riders

From the elegant cover design (with an explanation of how it was created on the back), to the conveniently separated subsections, it seems a great deal of effort was taken to make the book flow well, yet maintain a format where the reader can jump around without being confused.

The book starts by discussing some of the new features of LightWave 3D 5.5, and takes these features for a ride. While these first tutorials aren't advanced, they clearly illustrate some of the most powerful improvements with 5.5, such as the Knife tool, some of the new plug-ins such as Cyclist and Cel Shader, as well as the new Layout deformation plug-ins.

Something LightWave owners cannot avoid is dealing with installation and proper setup. The section on Installation and Setup is especially welcome! While we would rather be creative artists, we have to get our tools working first. Configuration files and video driver choices are a few of the issues covered here. This chapter, however, is meant for the beginner. More advanced coverage of Screamer-net and Render Farm setup would be a nice addition. The book is refreshing in that even the less obvious preparations, like how comfortable you feel at your workstation, office lighting, etc., are covered.

Any successful animator will tell you that a good grasp of color, motion, and lighting theory is extremely important. *Inside LightWave 3D* takes an artist-first approach to discussing LightWave. We begin our discussion on the various properties of lighting and color composition. Choosing proper colors for print and video work is also covered. Later on, the book talks in depth about texture mapping and reproducing lighting effects such as Radiosity. One of the tricks I liked the most was the use of displacement maps and texture mapping to make a mouse crawl underneath a rug: an amazingly simple solution to a complex problem. *Inside LightWave 3D* 5.5 is filled with similarly useful tricks.

One of the strongest aspects of *Inside LightWave 3D* is its in-depth discussion of modeling, especially character modeling. Even the long forgotten (but still extremely useful) splines are covered. Of course, MetaNURBs is covered in great detail. *Inside LightWave 3D* details MetaNURBs from an introductory to a very advanced level. It teaches advanced character and freeform body building secrets that can be used to create complex characters with ease. Even the more unpleasant aspects of NURBs, like proper triangulation, are not ignored. Not only are the tutorials excellent, but the objects on the accompanying CD are amazing and full of detail. Just looking at the sample objects served as a great source of inspiration. Among the many objects, the humanoid object was one of exceptional detail.

A well modeled character would go to waste without animation. In this area, the main focus is on character animation, because it is one of the most difficult type of animation to do. Being an expert character modeler/animator seems to be a valuable asset in recent times. *Inside LightWave 3D* takes you in that direction. Using the existing characters on the CD, the book discusses how to set up bones and animate them. An interesting technique involving splines is used to do this. These are not dancing logos; these are fully articulated, photoreal characters! While character animation can be a tedious process for beginners, the book contains plenty of diagrams and screenshots to help you along. However, I found that this section could have done a little more with gesture and motion theory.

(Continued on page 46)

NewTek Model Library

This CD Library of objects is aimed *specifically* at the Inspire 3D crowd, although the quality and selection of the 500+ objects will make this collection a hot product for the LightWave user as well. (The version being reviewed was a prerelease beta, so there may be some revisions before the final release.)

Library With A View

A nice feature of the CD is its Viewer program (Fig 1). The Viewer program has a nicely designed interface that allows you to scroll through the object directories and view a fully rendered image of each object (Fig 2). The picture appears in a quarter-screen window, which is actually about the size they would appear if placed on a video screen. While exploring all the objects with the Viewer program, techno-pop background music is playing. They have included a 'music off' button in the options panel, which I chose to utilize.

While at first the Viewer program seems like a nice addition, it quickly becomes clear that this is something that should be mandatory on all object collections. The truth is that most paying jobs in animation require rather mundane stock objects. The problem is that most common everyday objects come in a variety of shapes. For example, my workbench has four different types of hammers. Which hammer was my client envisioning when he asked for one?

Even if I have memorized all my stock objects, this instant reference saves me the time of finding the object, loading it into Layout, setting up a scene, and doing a quick render to find out if it is what I need. I can also show it to a client and get their approval without having to risk asking them to use their imagination to see what it looks like fully rendered.

Formats and Content

All of the objects on the CD are supplied in three different formats; .lwo objects for LightWave and Inspire, 3D Studio format, and DXF. The object are divided into these very useful categories:

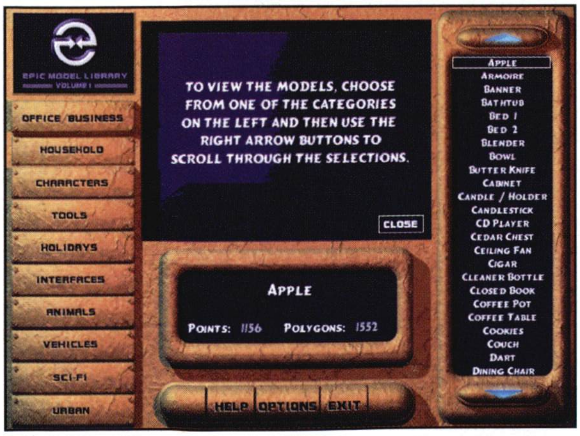
* Animals
* Characters
* Holidays
* Household
* Interfaces
* Office/Business
* Sci-Fi
* Tools
* Urban
* Vehicles

Without exception, every one of the objects in this collection is ready to go right out of the box. The surface textures for the most part are very minimal, with sparing use of image maps, but done well enough that they already look pretty good in the Open GL preview mode. While the surfaces of the objects are on the level of unpainted furniture, more advanced LightWave users will want to spend some time tweaking the textures to taste.

What Are They Good For?

As you can see from the categories listed, the objects cover many useful areas. It may be going too far to say you will never need to build an object yourself again, but it does cover many common as well as fanciful objects. Most of the common objects are pretty generic, yet still interesting visually. Many of the human and animal objects are of a rather cartoonish nature, which is fine if that is your need. However, using them as realistic creatures will not produce the results you desire. What I liked about the cartoon animals were that they had a lot of character and had a similar style. This means that you can put 'Super Dog', 'Fat Knight', and 'Jama Baby' all in the same scene, and they will look like they belong together.

(Continued on page 46)

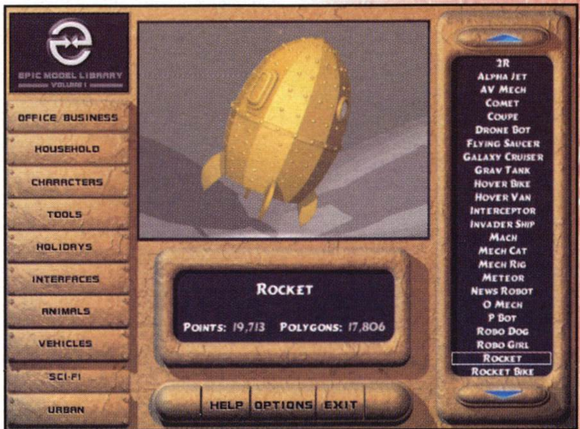

EPIC MODEL LIBRARY
VOLUME I
OFFICE/BUSINESS
HOUSEHOLD
CHARACTERS
TOOLS
HOLIDAYS
INTERFACES
ANIMALS
VEHICLES
SCI-FI
URBAN
CLOSE
HELP
OPTIONS
EXIT
Apple
Armoire
Banner
Bathtub
Bed 1
Bed 2
Blender
Bowl
Butter Knife
Cabinet
Candle / Holder
Candlestick
CD Player
Cedar Chest
Ceiling Fan
Cigar
Cleaner Bottle
Closed Book
Coffee Pot
Coffee Table
Cookies
Couch
Dart
Dining Chair

TO VIEW THE MODELS, CHOOSE FROM ONE OF THE CATEGORIES ON THE LEFT AND THEN USE THE RIGHT ARROW BUTTONS TO SCROLL THROUGH THE SELECTIONS.

Apple

Points: 1156 Polygons: 1552

Figure 1


EPIC MODEL LIBRARY
VOLUME I
OFFICE/BUSINESS
HOUSEHOLD
CHARACTERS
TOOLS
HOLIDAYS
INTERFACES
ANIMALS
VEHICLES
SCI-FI
URBAN
HELP
OPTIONS
EXIT
2R
Alpha Jet
AV Mech
Comet
Coupe
Drone Bot
Flying Saucer
Galaxy Cruiser
Grav Tank
Hover Bike
Hover Van
Interceptor
Invader Ship
Mach
Mech Cat
Mech Rig
Meteor
News Robot
O Mech
P Bot
Robo Dog
Robo Girl
Rocket
Rocket Bike

Rocket

Points: 19,713 Polygons: 17,806

Figure 2

Reviews You Can Use

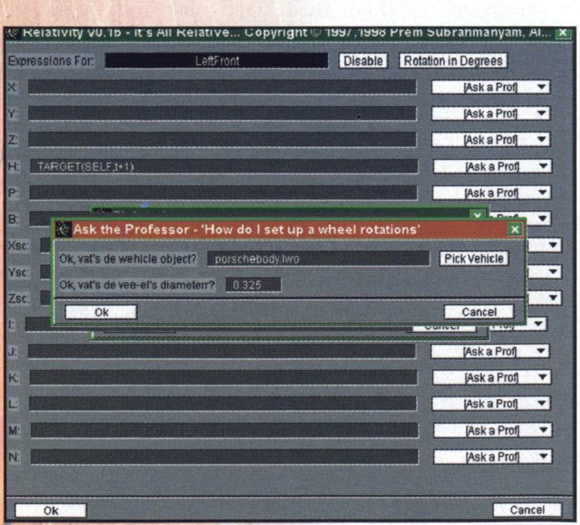
Relativity

Animations that used to take hours...are now set up perfectly in minutes. Relativity will have a revolutionary effect on the way you use LightWave!

Did you ever wonder what in the world you would use your high-school Algebra lessons for? You probably thought that when you graduated you would never see another Algebraic formula in your life...well, except for the famous one that has become more of a cliche than anything else. E=MC2 is just a small part of Einstein's Theory of Relativity, but it is the only part that most of us remember or even understand.

Well, luckily enough for the LightWave community, you do not need Einstein's intellect to use this incredible new motion plug-in, Relativity, from Prem Subrahmanyam Graphic Design. You do not even need to know anything about math to use Relativity, but it will help.

Imagine being able to apply a mathematical expression to the wheels of a car so that they rotate and turn correctly, regardless of how fast the car is moving or the direction it is turning. How about being able to make a missile launcher track a target and launch a missile with complete control over the missile's flight path, speed, and acceleration. Or how about being able to automate the movements of a character so that it runs, walks, and stands still depending on the motion path of the character! The possibilities are limitless, and you will be amazed by how easy it is to create these effects with Relativity!



Relativity v0.1b - It's All Relative... Copyright © 1997,1998 Prem Subrahmanyam, Al...

Expressions For: LeftFront

H: TARGET(SELF,t+1)

Ask the Professor - 'How do I set up a wheel rotations'

Ok, vat's de wehicle object? porschebody.lwo

Ok, vat's de vee-el's diameterr? 0.325

The Relativity motion plug-in interface has a simple layout which allows control over an object's position, rotation, and scale at any point in an animation. A "Professor" is available at the click of a button to help you create your mathematical expressions.

By using powerful mathematical expressions, Relativity gives you complete, independent control over the position, rotation, and scale channels of an item (any object, bone, light, or the camera itself), and relates that item to any other item in the scene at any point in time...you can even control time itself!

An easy-to-understand set of functions is used in the mathematical expression that determines exactly how an item is effected. All of the functions work in conjunction with each other, as well as LightWave's built-in mathematical functions.

Relativity has a powerful team of helpful "Professors" ready and waiting to assist you. Click a button and the Professor will ask you a series of questions to assist in the creation of an appropriate mathematical expression. Once you understand what each math function does, you will be able to easily write your own expressions, and then the fun REALLY begins!

At the time this article was being written, Relativity was in a limited beta- testing stage, but it was already surprisingly powerful. It took me less than a week to fully understand the different math functions and start writing my own mathematical expressions; this was without the aid of some of the more powerful Professors that are in development. By the time this plug-in is ready for full public release, I expect it to be one of the coolest and most powerful motion plug-ins available for LightWave. Relativity has already changed the way I work in LightWave! Animations that used to take hours to precisely keyframe by hand are now set up perfectly in minutes, and major changes can be accomplished just as quickly!

Just as Einstein's Theory of Relativity had a revolutionary effect on the scientific community, Prem Subrahmanyam's Relativity plug-in will have a revolutionary effect on the way you use LightWave!

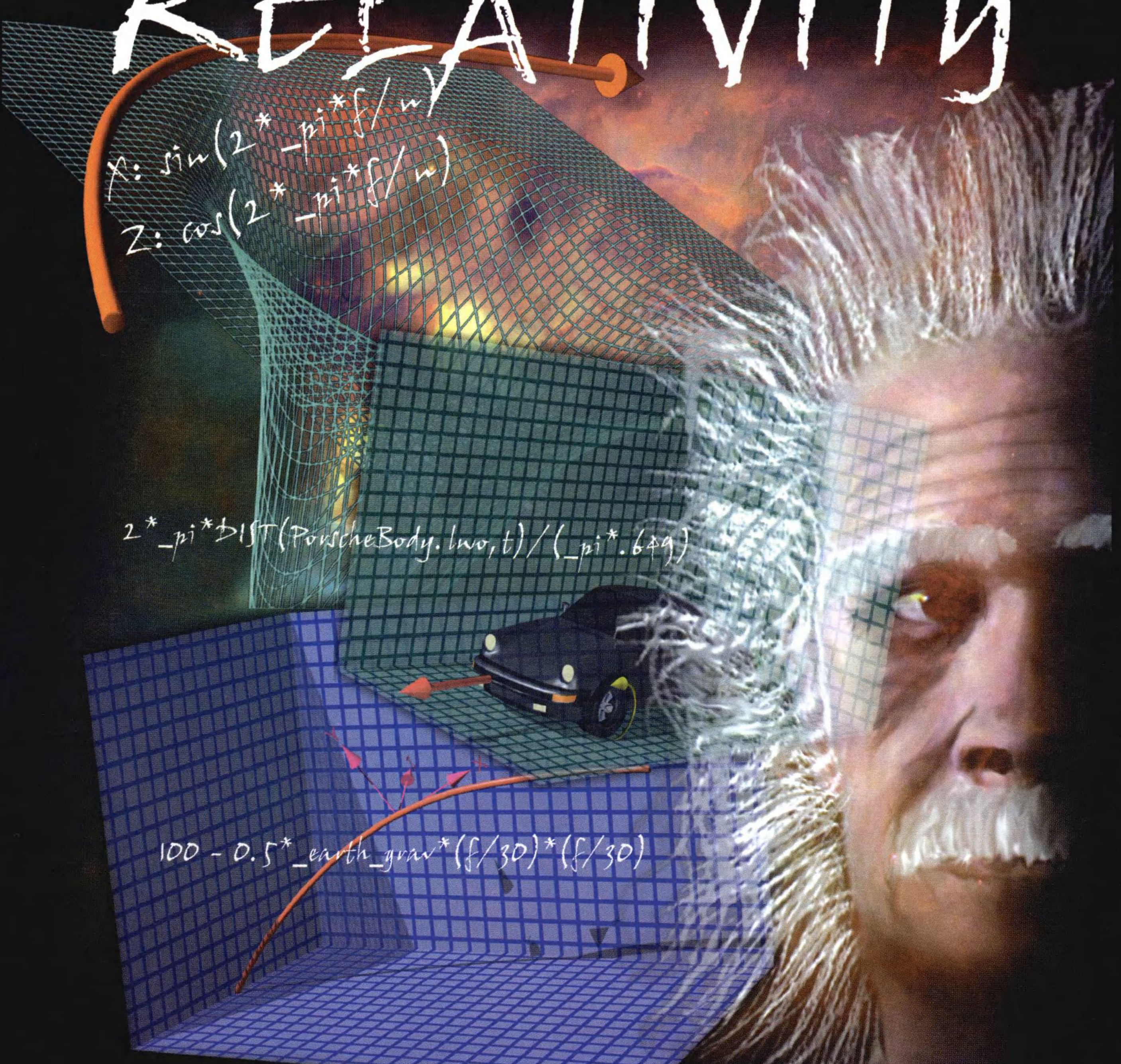
Relativity will be available by NAB 1998 with a suggested retail price of $180.00. Visit the NewTek booth at NAB for a mind-blowing demonstration!

For more information or to place an order, contact Prem Subrahmanyam via e-mail: at prem@dev.uol.com, or visit: http://www.noblestar.net/~prems/ppp.html. Phone: (904) 575-6051 between 11 AM and 6 PM EST only.

RELATIVITY

For more discussion of this topic, circle Reader Service #117

RELATIVITY


X: sin(2*_pi*f/n)
Z: cos(2*_pi*f/n)
2*_pi*DIST(PorscheBody.lwo,t)/(_pi*.649)
100 - 0.5*_earth_grav*(f/30)*(f/30)

The Revolutionary Expressions Plugin for Lightwave

Years ago, Albert Einstein formulated the Theory of Relativity, which had some strange implications. Space was related to time, time is related to space, indeed, space and time can be warped by gravity. These implications led to revolutionary developments in science.

Relativity, a new motion plugin for Lightwave, can launch a similar revolution in your animation work. Relate the motion channels of any item in Lightwave (camera, bones, objects, lights) to those of any other item at any point in time, using powerful mathematical formulas. Contract and extend time itself by relating it to an object's position. Create complex targeting relationships between objects. Even have objects snake along their own motion path. For character animation, relate walk cycles to the distance traveled by your character.

And what if you don't feel that you're a math genius? No problem...a room-full of helpful "Professors" can guide you in setting up complex relationships (like relating wheel rotation to a car's motion). This panel of luminaries will make setting up many useful and powerful expressions a breeze.

Whatever your expression needs, Relativity will help you solve the most important equation of all:

Time = Money.

Prem Subrahmanyam Graphic Design (850) 575-6051 http://www.noblestar.net/~prems/ppp.html

Reader Service #104

CD Content

DarkTree Texture Package

DarkTree Textures from Darkling Simulations is an exceptionally powerful tool that allows the digital artist to create amazing animated or static texture images, be they inspired by reality or fantasy. While the learning curve is somewhat steeper than some similar applications, DarkTree provides an awesome amount of flexibility and control over texture creation. Once the basic premise of DarkTree's algorithmic texture construction theory is absorbed, the imagination of the artist becomes the only constraining factor in achieving any desired results.

When DarkTree first loads, the user is presented with the initial DarkTree screen (Fig 1). This screen actually serves three purposes: as a navigator for organizing the library of textures, for initiating texture editing (either editing existing textures from the tree view or creating a new texture), and for rendered file output.

DarkTree Textures from Darkling Simulations is an exceptionally powerful tool

Texture Library

The tree view in the main screen, called the Texture Library, functions in the same way as the standard Windows convention, (i.e. folders and files), with the files being the algorithmic textures created in the DarkTree Editor. Double-clicking on a texture in the navigation tree brings up the editor screen, which we will get to later.

Bitmap Renderer/File Output

Once the texture has been created in the Editor and saved, it is dragged to the Bitmap Renderer for output. The renderer supports batch rendering, so you can create several textures in the editor and then render them overnight (algorithmic textures, by their nature, can take a while to render). File output formats include .bmp (16 and 24-bit), .jpg and .tga. DarkTree supports multiple naming formats for file output, as well as four levels of anti-aliasing (none, low, medium and high). Mapping type (such as planar, cylindrical and spherical) can be selected here as well as in the information section of the Editor. Since the textures are based on algorithms, they can be rendered at virtually any size and aspect ratio, making DarkTree Textures suitable for extra-high resolution work if needed.

Examine Window

Textures can be previewed at any time, either from the DarkTree Editor or the Main Screen, via the resizable Examine window. Supporting up to five image channels, the Examine window allows you to preview textures in progress, or see the result of multiple texture types when used together (i.e. as texture, bump, specularity and luminosity etc.) to create a complex surface in your favorite 3D application. The Examine window provides up to six views of your texture: front, back, top, bottom, left and right.

DarkTree Editor

The DarkTree Editor functions under a hierarchical tree-branch/parent-child metaphor, where the texture (the tree) gets its look from the attributes (parameters) of its root/parent as affected by the parameters of its branches/children. This is achieved by the use of components, each of which has parameters. Components are placed in sockets, and applicable parameters are "linked" together. When a component is dragged to a socket, a requester pops up asking what function the component performs: bump, color or percent.

DarkTree includes around 70 components, from the basics like bricks, checkers, dots and stripes, to complex mathematical generators such as sine, sawtooth and square waves and fractal noise. It also has gradient components, deformation components (stretch, turbulence, warp and others), process components (darken, lighten, add, subtract etc.), natural components (clouds, lumps, venation) and a host of others. External images are brought into the DarkTree environment via the Image and/or Image_Sequence components. DarkTree comes with an extensive library of pre-defined textures which you can modify, use as is or use to learn the different ways components and parameters can be combined to create virtually any effect imaginable.

A simple example: the "Bricks" component is dragged to the root socket of a new tree and established as a color component. The Bricks component has four linkable parameters: mortar_width, blend_function, brick and mortar. (bricklinks.bmp).

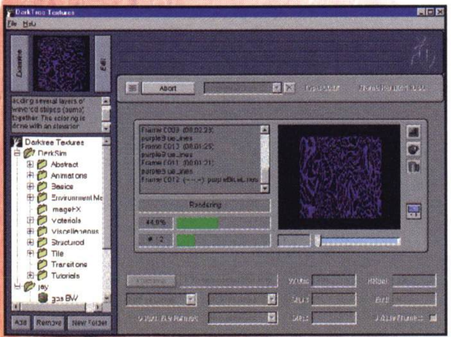

DarkTree Textures
File Help
Examine
Edit
Abort
Rendering
Darktree Textures
DarkSim
Abstract
Animations
Basics
Environment Ma
ImageFX
Materials
Miscellaneous
Structured
Tile
Transitions
Tutorials
jay
Add
Remove
New Folder

Figure 1

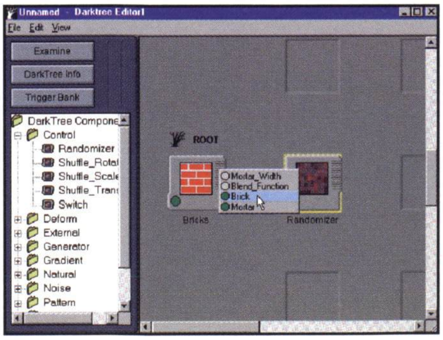

Unnamed - Darktree Editor1
File Edit View
Examine
DarkTree Info
Trigger Bank
DarkTree Compone
Control
Randomizer
Shuffle_Rotat
Shuffle_Scale
Shuffle_Trans
Switch
Deform
External
Generator
Gradient
Natural
Noise
Pattern
ROOT
Mortar_Width
Blend_Function
Brick
Mortar
Bricks
Randomizer

Figure 2

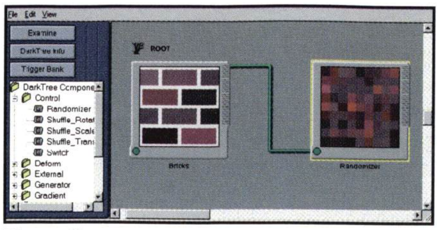

File Edit View
Examine
DarkTree Info
Trigger Bank
DarkTree Compone
Control
Randomizer
Shuffle_Rotat
Shuffle_Scale
Shuffle_Trans
Switch
Deform
External
Generator
Gradient
ROOT
Bricks
Randomizer

Figure 3

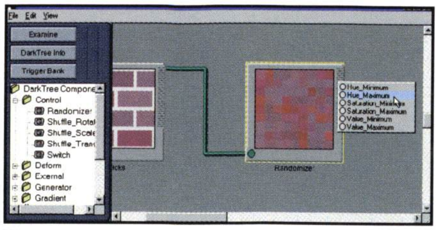

File Edit View
Examine
DarkTree Info
Trigger Bank
DarkTree Compone
Control
Randomizer
Shuffle_Rotat
Shuffle_Scale
Shuffle_Trans
Switch
Deform
External
Generator
Gradient
Hue_Minimum
Hue_Maximum
Saturation_Minimum
Saturation_Maximum
Value_Minimum
Value_Maximum
Randomizer

Figure 4

"Randomizer" component is then dragged onto the tree, set in a socket next to the Bricks component, and also established as a color component. By linking the "bricks" parameter of the Bricks component (Fig 2) to the Randomizer component, the coloring of each brick is subsequently randomized (Fig 3). By double-clicking on the Randomizer component, its parameters can be adjusted. While adjusting the values of the parameters, the effect of the new values is shown in the component editor preview window.

This window can show the effect on the root, or just the component itself. Note that the parameters on the component edit screen are color-coded, indicating that these are linkable as well. This can be seen by looking at the parameters on the link pad of the Randomizer component (Fig 4). Parameters are color-coded, indicating whether they are a color(green), percent(silver) or a bump(blue) parameter, and can only be linked to child components of the same type. This means that if you have the Bricks component as the root (established as a color root texture) and want to change the colors of the bricks, you must link the "brick" parameter (a color parameter) to a component that has been placed as a color component.

Be it known that DarkTree is not merely for creating fancy brick patterns. It is especially powerful at creating realistic and natural-looking gaseous vapor, water, fire and any other ethereal type textures (fire.bmp, aquafria.bmp, shown at right- included with DarkTree) as well as structured pattern-type textures. But the real strength lies in the fact that DarkTree Textures provides an intuitive and logical way of creating incredible-looking, animateable textures. DarkTree takes care of making the textures seamless, and also makes it easy to create looping animated textures. When multiple animated textures are used, for example a displacement map texture map and a transparency map, creating realistic explosions, fire, or water is a snap.

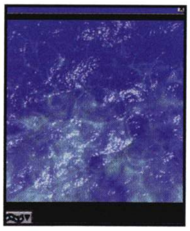
Conclusion

DarkTree Textures v1.0 is a serious and professional tool that, when wielded by an artist with a moderate level of technical acumen, can deliver extraordinary results. The included documentation is applicable, thorough and abundant, featuring a detailed discussion about the underlying theory behind algorithmic textures, in-depth descriptions about the capabilities and functions of DarkTree and the included components, and pertinent tutorials covering all of the concepts addressed therein.

Application: DarkTree Textures v1.0
Company: Darkling Simulations
Contact: Skyler Swanson
http://www.darksim.com
Platforms: Windows 95/NT
SRP: $270

AUTHOR PROFILE

Jay Watson is the president of the Helion Design, a multimedia/video company located in Orange County, CA. http://www.helion.com



aquafria.bmp

For more discussion of this topic, circle Reader Service #120

Preview
Issue #7 May/June

Theme: Hardware Considerations

New column: Video Out
Why LightWavers should know about the DV standard

Firewire DPS Edit bay vs. the Miro DC30+

Product Reviews: Vertitilies, Thor, Particle Storm 2.0

The Five Minute Series: Making Blueprints with LightWave

Programmer's Hideaway: Beginning LScript

Learning Curve:
New Tricks with Lens Flares
Making Blueprints
Effective RealiTools

Special Hardware Focus: Putting together the Best LightWave system for under $3000

Reviews You Can Use

Motion Pack Plug-In

There are some things that if you need, and if it is available, you have to have it and price is no object. This applies to 1/5 of NewTek's collection of motion plug-ins. There are five plug-ins in the package, four of which are useful but not indispensable, and one plug-in that could easily fetch twice the price all by itself. Let's quickly go through the minor players before we concentrate on the star. In order of the best first, they are:

Follower

One object can 'chase' another object. Very similar to saving out the motion path of the first object and then applying it with an offset to a second object. Using Follower is much more efficient because changes can be made to the first object that will affect the string of objects chasing it. Obvious applications are railroad cars following the engine, or text flying in a flock around objects. Variations and offsets can be applied to the Following object so that the Chasing object could actually be ahead of the leader. A nice little utility but with fairly limited applications; handy for quick logos and such.

Gravity

You can keyframe an object to a path, specify parameters of the bounce, and the object will make a bouncing motion along that path. There is no collision detection or squash and stretch unless you add it yourself. A nice little utility but with fairly limited applications; handy for quick logos and such.

Oscillator

Oscillator applies a dampening effect to any ping-pong type motion. The amount of damping can be offset and varied. If no damping is applied it is exactly the same as selecting 'Repeat' in the motion graph panel. A nice little utility but with fairly limited applications.

Serpent

You keyframe a path and your object conforms to the path as it moves along it. It is easy to use but it is very slow, especially if the snaking object is selected. There are at least two commercial plug-ins that do this already, and Issue 5 of *LightWavin' Magazine* explained how to get the same effect with no plug-ins at all.

All in all, Motion Pack is a nice collection of more or less useful plug-ins that would be very handy to the average LightWave user, but it is not meant for high-end, professional applications. It just seems odd to me that these are far less powerful than some of the plug-ins already included with LightWave.

Five tools for smooth, moving animation
MOTION PACK
PLUG-INS

Now we come to the real reason to buy the Motion Pack. One plug-in that instantly takes LightWave to a new era of possibilities is Cyclist.

Cyclist: Welcome to the Next Level

Basically, Cyclist lets you key action to motion. You can key the rotation of a wheel to the distance traveled over the path.

Anyone who has had the task of trying to match a vehicle's wheel rotation to the distance it travels appreciates how hard it can be, particularly when the vehicle changes velocity. Any repeatable motion can be applied to Cyclist. This includes things like people rowing a boat, a steam engine puffing steam, or a single complete WALK CYCLE!

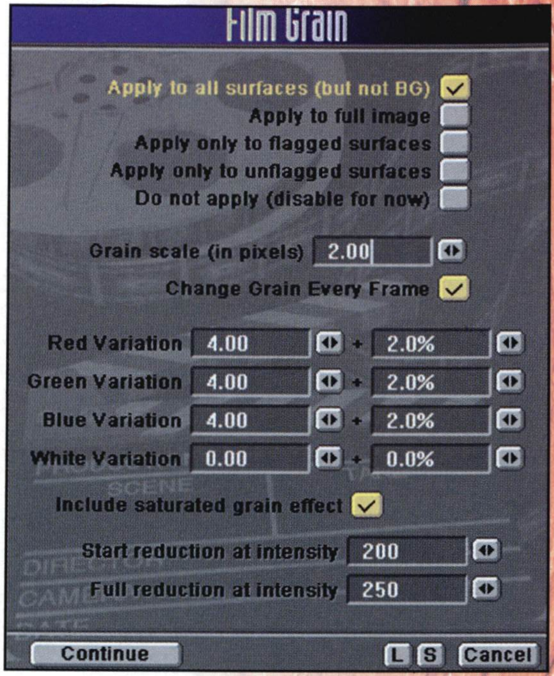
Remember when you saw Jurassic Park and you watched in awe as a herd of galimimus ran over the hills? Your automatic thought was, "I could do that if I had the time to keyframe each and every step along the path and make sure I vary the velocity to make it look natural and keep the feet nailed to the ground and clone the object and spend weeks tweaking the second path and vary its movements....." Not quite an impossible task, but a very, very daunting one.

I will say that 2 weeks after I got the motion pack CD, I created my own herd of running galimimuses. I did not exactly plan it; I was still learning how the plug-in worked, but I decided to apply it to the galimimus object from the Dinosaur CD. I was noticing how much better full time IK was working with version 5.5 and how wonderfully it integrated with Cyclist. Once I had a single perfect walk cycle I applied Cyclist. I have seen computers do incredible things, but watching my dinosaur follow the path, hesitate, and BACK UP, it was like I was seeing something magical.

After a while I saved it all and did a load from scene to give me two walking objects. By just scooting over the motion path I could have one dinosaur walking, the second come up to it, hesitate, let the other pass, and then continue on its way. The week before I got Cyclist I would not have even quoted on an animation job that would require that level of complexity. Before I was done I had a herd of 20 galimimuses running in a natural flocking motion in one side of the screen and out the other. An AVI is included on the cover CD. Keep in mind that this was really my first major animation using Cyclist and I had not actually planned to do it. In fact, because of the detail of the mod-

(Continued on page 46)

Film Grain Plug-In



Film Grain

Apply to all surfaces (but not BG) ☑
Apply to full image ☐
Apply only to flagged surfaces ☐
Apply only to unflagged surfaces ☐
Do not apply (disable for now) ☐

Grain scale (in pixels) 2.00
Change Grain Every Frame ☑

Red Variation 4.00 + 2.0%
Green Variation 4.00 + 2.0%
Blue Variation 4.00 + 2.0%
White Variation 0.00 + 0.0%

Include saturated grain effect ☑

Start reduction at intensity 200
Full reduction at intensity 250

Perfection Can Be A Curse

The images created by LightWave can be so pure that the pristine nature gives away their digital origin. The unnatural clarity often gives a sense of the image being 'artificial'. This is a problem when compositing LightWave objects onto background images which have been shot on film. Film is not a visually perfect recording medium; a "grain" can be seen with the naked eye. Yet for some reason we perceive filmed images as somehow being more natural. The grain results from the chemical processes necessary to fix images onto the celluloid. The finer the grain, the sharper the resulting image.

Over the years the technology of film has been steadily improving. Though we are not normally consciously aware of it, most people can make pretty accurate estimates as to the date (and budget) of a film after seeing just a few frames, mostly by the coarseness of the grain. We can also tell when a single frame is frozen, as opposed to filming a scene that is absolutely still, because of a situation called 'grain lock'. We need a certain amount of background noise, randomly generated color variation to the frames, to assure us that the film is still being run through the projector. In order to make LightWave generated images fit more comfortable on top of backgrounds originating of film, Newtek has released the Film Grain plug-in.

The Film Grain Plug-in was developed by Steve Worley, and effectively recreates the grain of film. A lot of thought and study was put into this deceptively complex process, aided no doubt by feedback from the Hollywood houses who first requested this ability.

Film grain can be applied to all surfaces, selectively to individual surfaces, or to the whole image including background and foreground. This pretty much covers all possible combinations. It can also be temporarily turned off for checking the non-grained image without losing the current settings. Settings can be saved and loaded, a useful feature for projects with multiple shots on the same film stock.

The main controls for achieving the proper look for the grain is Grain Scale and Grain Appearance. Grain Scale is the size of the blotches and defaults to 2 pixels. The documentation says that this is rarely changed. The 2 pixel default may be correct when working in film resolution for output back to film, but for compositing LightWave work over film that has been transferred to video for distribution on video I have found that settings between 1.05 and 1.20 work much better.

Grain Appearance sets the strength of a single grain and how much it may vary in color strength. The level of grain may increase with the brightness of the image, and this reaction can also be emulated by setting the variation setting. The grain amount and variation can be set individually for red, green, and blue. This is important because film is often color corrected during developing or may have an uneven color response due to the film stock or colored filters used on the camera. As a bonus, we also have the ability to create white colored grain, perfectly suited for black and white film.

The final appearance setting is called Grain Decay. The documentation does a much better job of explaining the process than I could do here. Essentially, it is a method of duplicating the way that the grain levels fall off during certain brightness levels. Grain matching at this level is probably necessary for IMAX theaters, but the 98% of us who will be using it for straight-to-video fare will have a hard time even seeing the difference that Grain Decay makes.

The Documentation

Compared to computer animation, actual filmmaking is an ancient art form. The documentation included as an html help file does a very good job of explaining the processes involved in duplicating film grain. It includes just enough explanation about why grain exists so we understand why certain controls exist on the plug-in panel. For computer animators with no experience in film (most of us) it may take several read-throughs of the documentation for them all to make sense, but mostly because it is such a completely different medium than we are used to.

What More Could You Ask For?

The Film Grain plug-in does the job it was intended to do perfectly, and of course we want more. The settings on the plug-in panel are of the 'set and forget' variety, meaning once you get the look dialed in you leave the settings as they are for the duration of the rendering. What would be nice would be the option of using a reference object to vary the levels over the course of the animation. This may not happen to real grain, but the ability to keyframe color variation and grain size sounds like a useful thing to be able to do.

The only real problem with the package is an unfortunate omission on the disk. The documentation refers to setting examples being included that could be loaded from the disk, but I was unable to find them.

Real World Uses

Film Grain's primary purpose is to help with the seamless compositing of LightWave generated images with footage shot on film, but it can do more. On a recent project it was required that I paint out a person for the second half of the shot (she had just been eaten by a LightWave monster). It was not too bad because she was standing in front of a static

(Continued on page 46)

(NewTek Library Review, Continued from page 39)

A particularly interesting directory is Interfaces: some very nicely laid out Interface screens in 3D just waiting to be applied to an application. Along with the Interface screen is a sub-directory of Icons, which are essentially buttons and arrows to place on top of the Interface. More experienced modelers can easily modify these interface objects to make 'button up' and 'button down' versions. Expect to see these interfaces popping up on web pages very soon.

What Is Missing?

What I do think should be included are some sample scenes. Novice users often gain a better understanding by examining a preconstructed scene. Some basic street scenes, office setups, or holiday scenes would be a great time saver when caught in a pinch and you need to throw something together for a presentation that literally is minutes away. Some scenes containing walk cycles for aliens, knights, and animals would be a godsend to people just getting started in animation; and some of the character objects are just screaming to be boned. A few morph targets would also be welcome, such as some different expressions for the characters' faces.

For Inspire users just starting in 3D animation, this CD is an excellent foundation with which to start building your catalog of resources. For the experienced LightWave user, it will be a valuable addition to your library of stock objects. Though the characters like 'Mega Stud' and 'Adventure Girl' are great objects, I think the objects that will prove most valuable will be things like 'trash can', 'road sign', and 'paperclip'. New users would spend too much time trying to build these basic objects, and expert users will want to spend their time on other .

For more information, visit NewTek's web site at www.newtek.com

For more discussion of this topic, circle Reader Service #119

(Film Grain Plug-in Review, Continued from page 45)

background, but rather then painting her out of all those frames I replaced the background with a single frame: a clean plate without the girl, and added film grain to it. It was wonderful how perfectly it worked!

Film Grain can be also added to completely CGI scenes. It gives them a little seasoning, kind of like adding reverb to an audio mix. By now you are probably thinking about using LightWave to give video footage a film look. That is a topic for another time, but adding a slight amount of film grain gets you more than half the way there.

Though none of them are as accurate and versatile as this plug-in, a few pieces of software have recently turned up that do some of the things previously only available with Film Grain. Look forward to a future article that explores the marrying of film and LightWave.

If you work with film, you need Film Grain, case closed, even if the final output is video tape. If you never touch film, you may still want to consider it to give your CGI a different yet familiar look, and any edge is important in a competitive field.

For more discussion of this topic, circle Reader Service #121

(Motion Pack Review, Continued from page 44)

els, it was necessary for me to render it in several passes and composite them together due to memory considerations.

A Suspicious Absence

The biggest drawback of this package is the lack of support material. Each plug-in is given a brief description that is just about adequate, and one or two tutorials. The Cyclist's examples are a wiggling fish and a unicycle moving in a straight line. 99% of this CD is empty. There is no excuse for not including a couple of examples of Cyclist being applied to walk cycles.

Conclusion

Cyclist is a major animation tool that is not getting near enough attention. If you want to do any thing involving characters, vehicles, or real world interaction, you need Cyclist. If you want to break into movies, if you want to make your own movies, then Cyclist is a tool to get you there. I think it was best summed up by a friend's reaction when I showed him the completed animation of my galimimus herd. "Looks like you're not just competing with the local studios anymore."

With my Cyclist equipped LightWave, that is exactly how I feel.

For more discussion of this topic, circle Reader Service #122

(Inside LightWave 3D Review, Continued from page 38)

Facial modeling and animation techniques are also an essential part of a CG characters' development. It also happens to be quite a challenging modeling task. An in-depth guide to Morph Gizmo, properly constructing facial structures using MetaNURBs, and creating morph targets are all discussed. One of the things I found most admirable is how easily one can create complex facial structures using the modeling techniques presented in the book.

Finally, a good character needs an environment to interact with. This is where compositing plays an increasingly important role. Compositing within LightWave is often an overlooked feature. However, *Inside LightWave 3D* makes it quite apparent that, with the right bag of tricks, some fairly impressive compositing can be achieved without the aid of external compositing software. The book follows a learn-by-doing approach, and I believe this is the easiest way to learn compositing, because you never know how an effect is going to turn out until you actually try it! Similar to the topic of compositing is video effects. This is of primary interest to those who use LightWave to create spots for video productions. Once again, all of the effects in the tutorials are done completely using only LightWave 5.5.

For the power user, the ability to custom-tailor LightWave's functionality has become more and more important. For the programmer types, there is a section on LScript written by the developer of Lscript, Bob Hood. This is probably the first useful set of tutorials on Lscript that I have ever seen. Best of all, the utilities that you develop are actually useful. There are also plenty of free plug-ins on the CD, as well as a demo of Surface Effectors to keep you busy.

LightWave 5.5 is deep and feature driven. We learned the basics in Dan Ablan's previous book, *The LightWave Power Guide*. Now, with *Inside LightWave 3D*, we can learn new and much more advanced techniques, and perhaps invent our own. It is truly a pleasure to see a book that goes beyond the basics and really focuses on creative and technically difficult challenges, yet at the same time keeps the beginner in mind. Whether you are just starting out with LightWave or you are an expert animator, *Inside LightWave 3D* should definitely be a valued member of your athenaeum.

For more discussion of this topic, circle Reader Service #118

Planet of Goods

WE WILL TRY TO BEAT OR MATCH ANY ADVERTISED PRICE

Get a free magazine with every purchase!

Genesis vfx
SPECIAL EFFECTS PLUGIN
FOR LIGHTWAVE 3D
$315.00

NEW LOW PRICES!
OXYGEN 102: $375.00
OXYGEN 202: $840.00
OXYGEN 402: $1675.00

FRACTALIZER
ONLY $129.00!

MESH PAINT 3D
WIN 95/NT: $379.00
3D STUDIO RELEASE: $ 299.00
DEC ALPHA VERSION: $439.00
POWERMAC VERSION: $379.00

WORLD CONSTRUCTION SET 3
INTEL: $640.00

PARTICLE STORM UPGRADE
$189.00

RealiTools
DEC ALPHA: $210.00
INTEL: $129.00

VIDEOS

Cinematic Visual Effects	$39.95
Digital Cinematography	$39.95
Space Craft Design	$39.95
Space Craft Surfacing	$39.95
Advanced Features	$39.95
LightWave QuickStart	$39.95
Mastering Modeler	$39.95
Surfacing Techniques	$39.95
LW QuickStart 5.5	$39.95
LightWave Lab vol. 1	$39.95
LightWave Lab vol. 2	$35.00
Digital Fantasies	$19.95
Organic Mod. & Anim.	$38.95
Steamer	$38.95
Plugin Power 1	$38.95
Pyrotechniques	$34.95

BOOKS

Power FX	$12.50
PhotoReal FX	$22.00
LW Character Animation	$44.00
LW Power Guide	$43.50
Inside LightWave 3D 5.5	$39.95
The LightWave Book	$29.00

PLUGINS

Con Motion	$395.00
Motion Pack	$229.00
Lock & Key	$167.00
Scene Machine	$185.00
Shaderman	$138.00
Surface Effectors	$77.95
Stripper	$105.00
Fiber Factory	$124.00
Film Grain	$134.00
Vertelectric	$95.00
Particle Storm 2	$CALL

CD-ROMS

Lenny Vol. 1	$68.95
Lenny Vol. 2	$42.00
Plaid Companion	$24.00
Digital Museum	$105.00
Universal 3DCD	$115.00
Digital FX CD	$46.95
Interior Design	$127.00
The Wright Collection	$127.00
LightRom 4	$44.00
LightRom 5	$38.00
LightRom Gold	$28.00
Movie Models	$39.00
Cemellot Collection	$123.00
TutorVision 3DCD Vol. 1	$49.95
Dolphin Vol. 1&2	$169.00
Dinosaurs ROM	$125.00

1-800-POGTOWN
1-800-764-8696 (U.S. ORDERS ONLY)
(805)726-3545

VISIT OUR WEBSITE!
HTTP://WWW.PLANETOFGOODS.COM

Reader Service #106

Final Cut

A New Challenge

Taking over a magazine is no small undertaking. When contributors, owners, and Editor reside in different time zones, and sometimes different continents, geography becomes a factor. Luckily, DMG Publishing has agreed to allow me to remain in my Midwestern home rather then relocate to their offices in California to perform my duties as Editor. Granted, California is the place you ought to be if you want to be in the thick of the LightWave/Hollywood scene, but LightWave is bigger then that.

LightWave may have had a significant part in the most successful movie ever made, but it reaches much farther then that. LightWave has also been an aid to the little guy. Wedding videographers, industrial producers, local advertising agencies, and even ultra-low budget film makers have the same tool that was chosen to carry the most expensive movie of all time. How many other people out there think this is really, really cool?

The Average User

I believe the average LightWave user works on projects with considerably lower budgets than those found on major Hollywood pictures. I also believe that with LightWave, the average user can match, at least for a few frames, the best that Hollywood has to offer. And I believe that the average user wants to create things that will knock people's socks off.

Pardon Our Dust

Due to a situation of timing, once the sale of the magazine was a done deal and I was installed as Editor, we realized we had barely enough time to get our first issue out for the National Association of Broadcasters (NAB) convention. This is one of the most important conventions in the business, and would be a perfect place to start our new life. The down side was that not only did we have all the bother involved in taking over the magazine, we would have less time than normal to get an issue together.

So if this issue appears to a have a few rough edges, attribute it to our commitment to meet deadlines and our dedication to a fresh new look. We are trying out some new columns and ideas, some of which may not make it to the next issue. We are eagerly awaiting your feedback on what you want to see, what you need to know, and what it is that makes you put an issue on the encyclopedia shelf insteadof into the recycle bin.

What the Future Holds

We plan on giving you more tutorials, more inside information, and more reviews on everything related to LightWave. *Keyframe* will remain 100% dedicated to the LightWave community. Expect to see both advanced techniques, as well as 'bread & butter' type stuff.

The fact that LightWave can be so valuable on such a large economic scale is incredible. If the success of a product is based on the number of people who have used it to support themselves and their families doing something they loved, then LightWave may be the most successful product of all time. I am glad that I am one of those fortunate people and proud to be a part of it.

Patrik Beck

Patrik Beck
Editor, Keyframe Magazine

Get Free information about the Products and Services mentioned in this Issue!

Circle the number on the card that corresponds to the Reader Service Number printed on the advertisement or referenced in the article, mail the card, and receive the requested information.

2763 W. Ave L Suite 162
Lancaster, CA 93536
805-264-4991 Fax 805-264-4998
888-778-WAVE

○ YES! Begin my subscription to Keyframe Magazine immediately! ○ no

Issue 6 May/June 1998

100	101	102	103	104	105
106	107	108	109	110	111
112	115	116	117	118	119
120	121	122	123	124	125
126	127	128	129	130	131
132	133	134	135	136	137
138	139	140	141	142	143
144	145	146	147	148	159
150	151	152	153	154	155

Name____________________

Company Name____________________

Address____________________

City__________ State____ Zip____

Phone (_____)__________

Fax (_____)__________

Email____________________

Signature____________________

☐ USA 1 year (6 issues) with cover CD	$48.00
☐ USA 1 year (6 issues) no cover CD	$24.00
☐ Canada/Mexico 1 year w/ cover CD	$60.00
☐ Canada/Mexico 1 year no cover CD	$36.00
☐ All other countries 1 yr w/ cover CD	$78.00
☐ All other countries 1 yr no cover CD	$54.00

All amounts shown in US Dollars

☐ Charge my Visa or Master Card
Card #____________________ Exp.____
☐ Payment Enclosed ☐ Bill Me

2763 W. Ave L Suite 162
Lancaster, CA 93536
805-264-4991 Fax 805-264-4998
888-778-WAVE

○ YES! Begin my subscription to Keyframe Magazine immediately! ○ no

Issue 6 May/June 1998

100	101	102	103	104	105
106	107	108	109	110	111
112	115	116	117	118	119
120	121	122	123	124	125
126	127	128	129	130	131
132	133	134	135	136	137
138	139	140	141	142	143
144	145	146	147	148	159
150	151	152	153	154	155

Name____________________

Company Name____________________

Address____________________

City__________ State____ Zip____

Phone (_____)__________

Fax (_____)__________

Email____________________

Signature____________________

☐ USA 1 year (6 issues) with cover CD	$48.00
☐ USA 1 year (6 issues) no cover CD	$24.00
☐ Canada/Mexico 1 year w/ cover CD	$60.00
☐ Canada/Mexico 1 year no cover CD	$36.00
☐ All other countries 1 yr w/ cover CD	$78.00
☐ All other countries 1 yr no cover CD	$54.00

All amounts shown in US Dollars

☐ Charge my Visa or Master Card
Card #____________________ Exp.____
☐ Payment Enclosed ☐ Bill Me

PLACE
POSTAGE
HERE

KEYFRAME
MAGAZINE

DMG Publishing
2763 West Avenue L, Suite 162
Lancaster, CA 93536

PLACE
POSTAGE
HERE

KEYFRAME
MAGAZINE

DMG Publishing
2763 West Avenue L, Suite 162
Lancaster, CA 93536

Subscribe Now!

Don't miss an issue of Keyframe Magazine.

Packed full of useful tutorials, product reviews, news, and interesting articles.

And it's all focused Exclusively on LightWave 3D and Inspire 3D.

KEYFRAME MAGAZINE

2763 W. Ave. L Suite 162
Lancaster, CA 93536
805-264-4991 Fax 805-264-4889

888-778-WAVE

○ **YES!** I want to start my subscription to Keyframe Magazine!

☐ Charge my card. (Visa or MasterCard only)

Card # ____________________ Exp. ____

Name ____________________

Company Name ____________________

Address ____________________

City ____________ State ____ Zip ____

Phone (______) ____________________

Fax (______) ____________________

Email ____________________

Signature (Required) ____________________

☐ Payment Enclosed US Dollars only

☐ Bill Me

☐ USA 1 year (6 issues) with cover CD	$48.00	☐ USA 2 years (12 issues) with cover CD	$84.00
☐ USA 1 year (6 issues) no cover CD	$24.00	☐ USA 2 years (12 issues) no cover CD	$42.00
☐ Canada/Mexico 1 year w/ cover CD	$60.00	☐ Canada/Mexico 2 years w/ cover CD	$108.00
☐ Canada/Mexico 1 year no cover CD	$36.00	☐ Canada/Mexico 2 years no cover CD	$66.00
☐ All other countries 1 yr w/ cover CD	$78.00	☐ All other countries 2 years w/ cover CD	$144.00
☐ All other countries 1 yr no cover CD	$54.00	☐ All other countries 2 years no cover CD	$102.00

KEYFRAME MAGAZINE

2763 W. Ave. L Suite 162
Lancaster, CA 93536
805-264-4991 Fax 805-264-4889

888-778-WAVE

○ **YES!** I want to start my subscription to Keyframe Magazine!

☐ Charge my card. (Visa or MasterCard only)

Card # ____________________ Exp. ____

Name ____________________

Company Name ____________________

Address ____________________

City ____________ State ____ Zip ____

Phone (______) ____________________

Fax (______) ____________________

Email ____________________

Signature (Required) ____________________

☐ Payment Enclosed US Dollars only

☐ Bill Me

☐ USA 1 year (6 issues) with cover CD	$48.00	☐ USA 2 years (12 issues) with cover CD	$84.00
☐ USA 1 year (6 issues) no cover CD	$24.00	☐ USA 2 years (12 issues) no cover CD	$42.00
☐ Canada/Mexico 1 year w/ cover CD	$60.00	☐ Canada/Mexico 2 years w/ cover CD	$108.00
☐ Canada/Mexico 1 year no cover CD	$36.00	☐ Canada/Mexico 2 years no cover CD	$66.00
☐ All other countries 1 yr w/ cover CD	$78.00	☐ All other countries 2 years w/ cover CD	$144.00
☐ All other countries 1 yr no cover CD	$54.00	☐ All other countries 2 years no cover CD	$102.00

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 143 LANCASTER, CA

POSTAGE WILL BE PAID BY ADDRESSEE

KEYFRAME
MAGAZINE

DMG Publishing
2763 West Avenue L, Suite 162
Lancaster, CA 93536

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

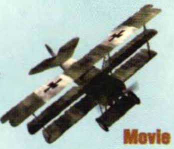
FIRST-CLASS MAIL PERMIT NO. 143 LANCASTER, CA

POSTAGE WILL BE PAID BY ADDRESSEE

KEYFRAME
MAGAZINE

DMG Publishing
2763 West Avenue L, Suite 162
Lancaster, CA 93536

Lightwave 5.5 (NT or PowerMac)§ • $1695
Hollywood's favorite Windows NT animation tool. New version includes loads of new features such as multi-processor support, Morph Gizmo, Volumetric rendering/lighting, Cel Shader, Direct 3D functions and more.



Movie Models • $79
CD-ROM with 21 high-resolution meshes of all manner of aircraft. Fully textured and ready to animate, in Lightwave format.
See samples on Lightwavin' CD!

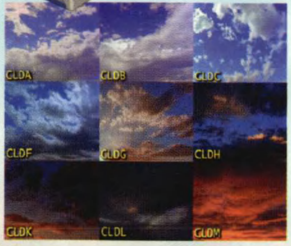
Pyromania! Pro

Pyromania! Pro
$279 *Specify Mac or PC.*
*An all-new collection of filmed pyrotechnics from VCE. Includes **30** high resolution sequences on 2 CDs with video catalog. Cinematic quality fire, explosions and more from 35mm film.*

Pyromania! Volume 1 & 2 • $179
The original Pyromania 1 & 2 CDs now available at an incredible new bundle price, including a video catalog. 28 segments, 640x480 or higher resolution. Specify Mac or PC.

Pyromania! Pro + Volume 1 & 2 • $379

See See samples on Lightwavin' CD!



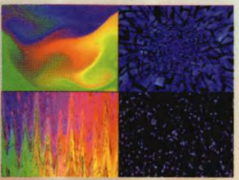
Colorado Altitudes • $159

Cirrus • $219
stunning timelapse cloud sequences, royalty free. Each 720x480 resolution, 150-300 frame sequence was scanned from film for rich color. PC/Mac CD-ROM with Video.
See samples on Lightwavin' CD!

ReelFire2
ReelExplosions1
ARTBEATS
ReelFire1
ReelExplosions2

ArtBeats ReelEffects
30 segments in the theme of fire and explosions on each Volume. Sequences at 720x486 or larger.

ReelFire 1 • $349
ReelFire 2 • $349
ReelExplosions 1 • $349
ReelExplosions 2 • $349



Multimedia Ingredients • $79
CD-ROM full of animated abstraction and music. Contains 45 unique, moving sequences, royalty-free.
See samples on Lightwavin' CD!

Flight Technologies 3D Workstations • $CALL for custom quote!

Plug-Ins for Lightwave

For Intel unless noted.
(I/A) = price is for both Intel and Alpha

Genesis	$340
Puppet Master	$310
Puppet Master (Alpha)	$360
Vertilectric (I/A)	$90
MacroForm (I/A)	$289
Lock & Key	$190
Lock & Key (Alpha)	$250
Impact	$390
Impact (alpha)	$490
Con Motion	$485
Particle Storm	$260
Particle Storm (Alpha)	$330
Particle Type (Intel)	$99
PointAt (I/A)	$65
Time Machine (I/A)	$105
Wobbler (I/A)	$69
Fiber Factory	$130
Fiber Factory (Alpha)	$180
Sparks	$190
Sparks (Alpha)	$230
WaveFilter	$170
Shaderman	$160
Shaderman (Alpha)	$249
Stripper	$149
Stripper (Alpha)	$219
Logo Wizard	$199
Relief Landscape	$199
Motion Pack	$260
Surface Pro	$79
Surface Effectors	$95
Scene Machine	$149
Digital Art Brush	$109
Extract Audio (I/A)	$69
WaveNet Pro	$CALL

Software

Lightwave 5.5 Upgrade	$459
After Effects 3.1	$590
AfterFX Prod.Bundle	$1390
Digital Fusion 2	$1890
Ultimatte	$1330
Adobe Premiere 4.2	$490
3D Studio Max 2 §	$CALL
4D Paint Rembrandt §	$439
4D paint Goya (Lite) §	$235
Fractal Detailer	$369
Poser 2	$179
Fractal Painter 5	$389
World Cons. Set 3	$670
DeBabelizer Pro	$499
Animotion Spaceball	$390

CD-ROMs

Cirrus	$219
Lightspd Dinosaurs	$139
Humanoid 2 LW	$179
Digital Museum	$129

Lightwave Videos

Quickstart	$45
Mastering Modeler	$45
Advanced Features	$45
Surfacing Techniques	$45
All 4 above	$165
Space Craft Modeling	$45
Space Craft Surfacing	$45
Digital Cinematography	$45
Cinematic Visual FX	$45
All 4 above	$165
Disp. Map, Morph, Bones	$45
Light & Cam. Techniques	$45
Pro Flying Logo Tech.	$45
Pyrotechniques	$45
Steamer	$45
Starting Your Anim. Biz	$55
Organic Modeling & Anim	$45
Plug-In Power	$45

Sales: M-F 8am - 6pm CT
FAX: 816-525-1594
e-mail: info@trinity3d.com
Int'l: 816-525-0103
International Orders Welcome (except where noted by §)
Note prices subject to change, call for current pricing on listed products.

TRINITY
ANIMATION & VISUAL EFFECTS

TRINITY
ANIMATION & VISUAL EFFECTS
676 SE Bayberry Lane, Suite 103B
Lee's Summit, MO 64063-4389
Don't see it here? Call for price!
Same day shipping before 4 pm
WWW: www.trinity3d.com

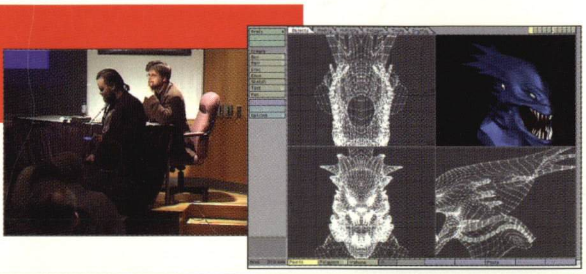
800-548-1578

800-548-1578

LENY 1998
LIGHTWAVE EXPO NEW YORK

LIGHTWAVE 3D EXHIBITS & SEMINARS AT LENY '98 SATURDAY MAY, 23 1998!

LEE & KEN STRANAHAN
90 MINUTES / $50 PER PERSON



Lee and Ken have been staples at *LENY* for the past two years. They're back again for LENY '98! The Stranahan brothers will wow you with killer LightWave techniques used in Hollywood's top productions.

METANURB MADNESS WITH MATT MCDONALD
90 MINUTES / $50 PER PERSON

Matt McDonald is the founder of the newly formed *Vision Scape Imaging* and *The Lab* 3D animation training center. Matt created the LightWave 5.5 content CD. You've also seen his company's work on TV, in video games and soon to be released feature films. Here's your chance to master metanurbs with the madman!

ENTERING THE DIGITAL DOMAIN WITH ERIC BARBA
90 MINUTES / $75 PER PERSON

NO VIDEO RECORDING PERMITTED IN THIS SEMINAR

Digital Domain's use of LightWave in films like *Titanic* and scores of award winning TV commercials is legendary in the industry. In this seminar, LightWave artist Eric Barba reveals the secrets of one of Hollywood's top visual effects studios.

DIGITAL DOMAIN

SHOW YOUR REEL TO A PRO!
DD RECRUITING WITH LAURENCE PLOTKIN

***Digital Domain's** head of recruiting will be reviewing attendee's reels, and talking about what he looks for in new hires. Don't miss your chance to get a foot in the door with Hollywood's LightWave powerhouse!*

780 THIRD AVENUE BETWEEN 48TH & 49TH STREETS NYC

- FREE ADMISSION TO THE EXHIBIT AREA!
- FREE TECHNOLOGY SEMINARS FROM TODAY'S TOP 3D COMPANIES!
- FREE ADMISSION TO EXCLUSIVE ACE MILES LIGHTWAVE SEMINAR!
- WIN LIGHTWAVE 3D 5.5!!! (SEE WEB PAGE FOR DETAILS)

NewTek digital INTERGRAPH COMPUTER SYSTEMS DIAMOND MULTIMEDIA

MetaCreations Dynamic Realities DIGITAL DOMAIN

msd
Multimedia System Design

(888) MSD-NYC1
WWW.MSD-NYC.COM